The Impact of Feedback on the Motivation of Software Engineers

Thesis

The Impact of Feedback on the Motivation of Software Engineers

A thesis submitted to The Open University
for the degree of
Doctor of Philosophy in Computing

by

Rien Sach

2013

ABSTRACT

This research investigates how feedback affects the motivation of software engineers and develops a model of feedback in software engineering. Motivation has been reported as having an impact on software engineers' productivity, the quality of the software they produce, the overall success of a software development project, and on the retention of software engineers. Findings from the last 30 years of research investigating motivation in software engineering have identified several factors that influence the motivation of software engineers**,** but the impact of each individual factor remains unclear. Feedback was identified as a factor affecting motivation by several studies investigating motivation in software engineering. Several theories of motivation exist which identify factors affecting motivation and models of how motivation is affected. Feedback is identified as a factor in four theories of motivation. In 2008 a systematic literature review identified that the majority of previous studies investigating motivation in software engineering were not grounded in motivation theory. This suggests that the majority of previous research investigating motivation in software engineering has not adequately considered theories of motivation and their relevance in software engineering.

This research explored the importance of feedback and the effect of the characteristics of feedback on the motivation of software engineers, collecting their thoughts, perceptions, reflections and reactions to feedback using a range of different research methods. The research began with a preliminary study investigating how software engineers perceived feedback, and if the characteristics they identified were comparable to those identified in other disciplines, notably clinical education. Further studies followed by investigating feedback in software engineering, the short-term impact of received feedback, and the effect of the 'source' and 'medium' feedback characteristics.

The findings of the preliminary study were that software engineers identified characteristics of feedback comparable to those found in clinical education, which gave a basis for further studies. Software engineers reported that feedback was the most common method of tracking their individual progress in a software project. A diary study collecting instances of feedback reported by software engineers found that positive feedback typically increased the engineers' job satisfaction, and that negative feedback typically led to a change in their behaviour.

Building on the earlier findings of this research, a scenario study and an online survey combining both scenarios and questions investigated the effect of the source and medium feedback characteristics. The findings of the four studies identified that the feedback recipient’s values and perceptions of the feedback source, and any preference they had to the medium used to send the feedback, affected the impact of received feedback. The findings suggested that the feedback software engineers report as the most valuable is not the same as feedback reported as having the most impact. The findings suggest that in software engineering, theories of motivation do not adequately consider the impact of the characteristics of feedback and the effect of different forms of feedback on motivation.

A model of feedback in software engineering was identified by combining the findings of the four empirical studies and relevant literature. The model captures how feedback is experienced by software engineers. Software engineers perceive the characteristics of the received feedback, which provides information that is used to make several assessments about the feedback. Each engineer’s individual value set influences their assessments, and their current state of mind / mood / emotions affect the engineer’s perceptions, assessments, and individual value set. The assessments of the feedback then result in the impact of the received feedback, which can have an effect on the engineer’s attitude, behaviour, motivation, performance, job satisfaction, and feelings.

ACKNOWLEDGMENTS

This research is the culmination of over three and a half years of hard work; however it would not have been possible and anywhere near as successful without the support, belief, and commitment from a range of different people. For this, I will always be indebted to you, and hopefully this small acknowledgement will go some way to representing how grateful I am to all of you for your time and support over the past years. In no particular order:

Thank you to Julius Davies, Katie Wilkie, Keerthi Thomas, and Peter Coles for helping with the piloting of multiple research methods; their time was and remains greatly appreciated. Dominic Mason proved to be the perfect ‘guinea pig’, giving up much of his free time to help me pilot all of the methods used during this research. His now newly-wed wife Ally Mason was very understanding in allowing Dom to help, for which they both are greatly appreciated.

My supervisors, Helen Sharp and Marian Petre; who had the initial belief in me a few years ago and gave me this opportunity, then pushed me from the start to be the best researcher and best academic I could be. Without their persistent comments, challenges, and encouragement I would have never finished this research.

Thank you to Simon Butler for willingly helping me find participants while at a conference I could not attend, and engaging with my research and providing insightful comments. Thank you to Clara Mancini for useful and knowledgeable advice on the use and implementation of participant-reported diaries.

Thank you to all of my fellow computing research students at the Open University, especially those who helped my development throughout my PhD by providing comments on my research, and helping me to better express my thoughts and findings. Thank you to Leonor Barroca and Hugh Robinson, who challenged me and instilled me (maybe unintentionally!) with the confidence to continue my studies during my probation mini-viva. Thank you to The Open University for funding this research and providing me with an environment and research structure that supported and facilitated my studies.

My family have seen the real ‘impact’ this research has had on me and my life. They have provided me with advice, with support, and most of all with their time both directly and indirectly. My grandparents, Jim and Joan Sach have always

encouraged me to ‘stick at it’ and have shown unreserved appreciation for the stress and strains of this research. My Nan, Daisy Pratt has, at all times, been there to help me with anything and everything I could possibly need, helping to fund my undergraduate degree, and at all times offering me anything she could. I genuinely believe this PhD means as much to my Nan as it does to me, and I am sure she would sell all her possessions to make sure I completed it if I asked her to.

My parents, John and Debbie Sach, have always pushed me to excel and succeed in life, providing support, both in words and financially, through my life and throughout my studies. I have probably been the most teenage-like 20-something year old in my grumpiness and shortened-temper during this research, and at all times my parents have supported me and my decisions, even ones that make their life more difficult. In particular, helping me to find my own ‘man’s best friend’ and looking after her while I went off all over the world during this research.

Last but by no means least, my girlfriend; Catrina Sherrin has had to put up with a relationship that has been anything but normal. A vanishing boyfriend gone for days at conferences, cancelled dinner plans due to pressing deadlines, and more recently having to pick up the slack around the house. Most of all, being able to put-up with someone spending all the time either doing their research, or talking about their research. Her commitment, resilience, and selflessness have ensured I’ve had all the time I needed to see this research through to completion.

Thank you all once more for your time, effort, and support throughout this research. It has been fundamental in enabling me to complete this research and achieve something I doubted I would be able to do.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PUBLICATIONS

The research reported in this thesis has contributed to the following publications:

R. Sach, M. Petre, and H. Sharp, "The use of MBTI in software engineering," 22nd Annual Psychology of Programming Interest Group 2010, 19-22 September 2010, Universidad Carlos III de Madrid [Sach *et al.* 2010].

R. Sach, H. Sharp, and M. Petre, “What makes software engineers go that extra mile?” 23rd Annual Psychology of Programming Interest Group 2011, 6-8 September 2011, York, UK [Sach *et al.* 2011a].

R. Sach, H. Sharp, and M. Petre, “Software Engineers’ Perceptions of Factors in Motivation,” 5th International Symposium on Empirical Software Engineering and Measurement, ESEM 2011, 22-23 September 2011, Banff, Alberta, Canada [Sach *et al.* 2011b].

R. Sach and M. Petre, “Feedback: How does it impact software engineers?” 5th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), ICSE 2012 Workshop, June 2nd, 2012, Zurich, Switzerland [Sach *et al.* 2012].

The listed publications included data presented in this thesis, as identified below:

Chapter 5 – The data collected from the motivation and de-motivation questions during semi-structured interviews with software engineers was presented in [Sach *et al.* 2011a] and [Sach *et al.* 2011b].

Chapter 6 – The findings from the analysis of a scenario study with software engineers was presented in [Sach *et al.* 2012].

GLOSSARY

Please note the information provided in the brackets identifies the context in which the term is used. Some of the identified terms (example: value) have multiple contexts in which they are used, and some terms are easier to understand if a context is provided.

Term	Definition
Attribute (of Characteristic):	Sub-characteristic of a characteristic of feedback, for example the experience of the source, or the polarity of the content.
Characteristic:	A feature of feedback, for example medium, source, and content.
Aspect:	Something identified in previous literature or during this research as being affected by feedback.
Feedback:	The transfer of information prompted by a form of stimulus. Feedback can be characterised by the values of ten characteristics.
Form (of feedback):	A variation of feedback defined and identified by the value of the feedback's characteristics.
Instance (of feedback):	A single occurrence of feedback.
Motivation:	Motivation, in this thesis, refers to the impetus and direction behind an individual's behaviour and/or actions. The definition of motivation is disputed. A more detailed discussion, including the presentation of different theories and definitions of motivation can be found in chapter 2.
Value (of feedback source):	If the source of the feedback is valued by the recipient. For example a recipient may place high value to feedback received from a colleague, but may not value feedback received from their line manager.
Value (of feedback content):	If the content of the feedback is valued by the recipient, for example a recipient may value feedback that has technical content, but may not value feedback discussing their behaviour.
Value (of feedback	A value assigned to a feedback characteristic, for example

characteristic):	email or note for medium, and colleague or manager for source.

CHAPTER 1

INTRODUCTION

1.1 PROBLEM OVERVIEW

Studies investigating the impact of motivation in software engineering have identified that motivation can affect the productivity of software engineers (Procaccino *et al.*, 2005), that it can affect the quality of produced software (Boehm, 1981), and that it is frequently reported as a cause of an unsuccessful software project (DeMarco and Lister, 1999). Dealing with unmotivated software engineers has been found to consume up to 60% of a project's budget (Abdel-Hamid, 1989). All of these findings relate to the study of motivation in a global computer software market valued at over $265 billion as of 2010 (Datamonitor, 2011), and expected to be valued at over $350 billion by 2015 (Datamonitor, 2011).

Several theories of motivation exist within the literature, each providing a slightly different interpretation of motivation. Broadly theories of motivation view motivation either at a single point in time, or as a combination of different events and factors occurring over time. The theories of motivation provide a suitable foundation for any study investigating motivation in software engineering, although the majority of previous research investigating motivation in software engineering has not been driven by theories of motivation. A review of theory use in studies investigating motivation in software engineering (Hall *et al.*, 2009) found that of all reviewed studies, less than half adequately used theories of motivation as an underpinning foundation to the study during the research. Often theories of motivation were either not used at all, or were used as a comparison to the findings of the study. While some studies did use theories of motivation adequately, many previous studies investigating motivation in software engineering did not, questioning the robustness of these studies.

Four theories of motivation identify feedback as a contributing factor. Job Characteristics Theory (Hackman and Oldham, 1976), Goal Setting Theory (Locke, 1968) and Achievement Theory (McClelland, 1961) argue that feedback is used to provide the recipient with knowledge of the outcome of their actions, and Hygiene theory (Herzberg, 1959) claims that 'verbal feedback' is the most common way an individual receives recognition for something he/she has done. Empirical studies of software engineers identify feedback as a factor affecting motivation (Couger 1989; Cheney 1984; LeDuc 1980; Wegge and Haslam 2005). However, this has only been the case when investigated through the use of a theory of motivation that already identifies feedback as a factor. Studies investigating motivation in software engineering and theories of motivation to the best of our knowledge do not consider the effect of different forms of feedback, contextual factors, and the feedback recipient's individual characteristics, values and preferences.

The lack of previous research investigating feedback, its definition, and the role that it plays in affecting the motivation of software engineers is a significant gap in the knowledge of motivation in software engineering, and how feedback can affect the motivation of software engineers. Investigating this gap will provide more knowledge for the body of research on motivation in software engineering, what the effect of feedback is on software engineers motivation, will inform software engineers and their managers of the importance that feedback has in software development environments, and will enrich theories of motivation with an enhanced definition of feedback and an improved understanding of how feedback can affect motivation in software engineering.

This thesis focuses on one factor, feedback, and investigates the impact feedback can have on the motivation of software engineers. Feedback is a factor identified in four theories of motivation as affecting an individual's motivation. In studies of software engineers, feedback has been identified as a factor affecting motivation, but no prior studies were found that investigated feedback individually, or identified the effect of different forms of feedback. Feedback remains an unknown factor in software engineering, with little recent research focused on better understanding feedback. This research focuses on software engineers, and investigates how feedback can affect their motivation in software engineering environments.

This research included four empirical studies with experienced software engineers. Using multiple research methods combining qualitative and quantitative research methods, this research investigated an underexplored field requiring a flexible and adaptive research approach. Overall, the empirical research started by using a qualitative approach based on interviews and diary studies, and progressed to a quantitative experimental survey, with each of the four empirical studies building on the findings of the previous study. The initial qualitative approach was suitable for a field requiring a broader investigation due to limited current knowledge, and the quantitative experimental survey was appropriate to investigate the generalizability of the findings from the first three studies.

In chapter 3, a qualitative study was designed to investigate how software engineers discuss feedback. A semi-structured interview collected qualitative data that was analysed qualitatively to identify emergent themes. Building on the findings presented in chapter 3, chapter 5 investigates the different forms of feedback found in one software engineering environment, combining quantitative and qualitative research approaches. Quantitative data was collected identifying the values of feedback characteristics through an online diary study and was analysed both quantitatively by counting the number of instances of feedback types, and qualitatively by identifying emergent themes reported by participants. Qualitative data was collected through a semi-structured interview investigating what motivates software engineers and what feedback exists, and was qualitatively analysed by identifying emergent themes reported by the participants.

In chapter 6 a scenario-based study is presented that builds on the findings of chapters 3 and 5 by investigating the impact of received feedback, and how the values of feedback characteristics can change the impact of received feedback. In chapter 6 quantitative and qualitative research data is presented. Participants first indicated the impact of feedback in the form of a tick-box, which provided quantitative data that was then counted. The participants discussed their responses during a follow-up semi-structured interview, providing qualitative data that was analysed with a qualitative approach identifying emergent themes.

Chapter 7 presents the findings of an experimental survey collecting quantitative data investigating how the medium, the source, and the content polarity affects the impact of received feedback, building on the findings of the previous empirical studies presented in chapters 3, 5 and 6. The experimental survey presented in chapter 7 was analysed quantitatively, identifying the number of respondents indicating each answer. The collected data was analysed, and emergent patterns identified. Confidence intervals were calculated for the data to reduce the likelihood that any patterns were by chance.

The progression of the research was data-driven, with each empirical study building on the previous study's findings, refining the focus of the investigation where relevant based on the knowledge gained from each previous study. This research focused on the thoughts, reflections, perceptions and reactions of software engineers that were collected during the empirical studies.

1.2 RESEARCH FOCUS

Feedback, in this research, is defined as 'The transfer of information prompted by a form of stimulus. Feedback can be characterised by the values of ten characteristics'. Feedback is a factor identified as affecting motivation. Motivation can be viewed as being intrinsic or extrinsic: Herzberg (1987: 14) defines intrinsic motivation as *"a function of growth from getting intrinsic rewards out of interesting and challenging work" (Herzberg, 1987: 14)* and extrinsic motivation as *"a function of fear of punishment or failure to get extrinsic rewards." (Herzberg, 1987: 14)*. Feedback is an extrinsic factor which from a review of literature and from the findings of this research has been found to influence both intrinsic and extrinsic aspects of software engineers' motivation.

This research uncovers and establishes characteristics of feedback as identified and discussed by software engineers working in software development environments. Ten feedback characteristics, and their values, are investigated, focusing specifically on the source: who or what provides the feedback, the content polarity: whether the feedback is positive, negative, or neutral, and the medium: how the feedback is received.

Throughout the thesis, the overall research question is:

How does feedback affect the motivation of software engineers?

This overarching research question covers the scope of this thesis, specifically investigating how feedback is identified and discussed by software engineers, if literature in this discipline and other disciplines can help establish a definition of feedback in software engineering, and then continuing to investigate how feedback, and different forms of feedback as identified by the values of feedback characteristics, can affect different aspects of software engineers. Specifically, the focus of this research is the effect of feedback in software engineering environments, on software engineers. The research takes an empirical approach, eliciting data from software engineers using several research methods, and at all times focuses on how software engineers discuss and report feedback, and the impact that software engineers believe feedback can have on them.

From the findings of the research the source, the medium, and the content polarity are shown to have the most effect on changing the impact of received feedback, and as the research progresses these characteristics are investigated further. Overall, this research provides insight into how feedback is experienced by software engineers, and develops a model of feedback in software engineering that includes both relevant literature and the findings of this research to provide a contribution to knowledge in software engineering.

As the research progressed it became apparent that investigating the overall research question would lead to the emergence of other more focused research questions. In the following sections the focused research questions are presented, and the chapters of this thesis that investigated each research question are identified.

The literature review identified that there was insufficient research investigating feedback in software engineering, and no research was found that focused solely on feedback in software engineering. The literature review prompted the following three research questions:

> **Q1 – Do software engineers report feedback as commonly occurring in software engineering environments?**
>
> **Q2 – How is feedback defined by software engineers?**
>
> **Q3 – What are the characteristics of feedback in software engineering environments?**

The Feedback Scoping Study (chapter 3) investigated questions one to three. From the findings of the preliminary study, and the investigation of the first three focused research questions, additional research questions emerged:

> **Q4 – What forms of feedback do software engineers report receiving?**
>
> **Q5 – What is the initial effect of received feedback?**

Q6 – What is the delayed effect of received feedback?

The Feedback in One Software Engineering Environment Study (chapter 5) investigated questions four to six. The findings of the Feedback in One Software Engineering Environment Study provided answers to questions four to six, but by answering question five, further questions emerged about the effect of the ‘medium’ and ‘source’ feedback characteristics. From the findings of the Feedback in One Software Engineering Environment Study, the following research questions emerged:

Q7 – Are there any feedback characteristics that change the impact of received feedback?

Q8 – What is the effect of a change in feedback medium?

Q9 – What is the effect of a change in feedback source?

Q10 – Why does the value of a feedback characteristic change the impact of received feedback?

The Short-Term Impact of Received Feedback study (chapter 6) investigated questions seven to ten and the Effect of ‘Source’ and ‘Medium’ Feedback Characteristics Study (chapter 7) investigated questions seven and nine, identifying the effect that the value of the ‘source’ and ‘medium’ feedback characteristics could have.

The findings from the empirical studies provided insight into feedback in software engineering, addressing the overall research question and the ten focused research questions. These findings led to two analytic research questions, which used and applied the findings of the empirical studies:

Q11 – How do the findings of this research compare to theories of motivation that identify feedback as a factor?

Q12 – What does a model of feedback in software engineering look like?

These research questions were addressed separately. The Discussion (chapter 9) compares the findings of this research with relevant theories of motivation that identify feedback as a factor, addressing question eleven. A Model of Feedback in Software Engineering (chapter 8) brings together all of the relevant findings from this research and relevant literature to identify a model of feedback in software engineering, addressing question twelve.

1.3 THESIS STRUCTURE

The structure of the rest of the thesis is presented in the following paragraphs, and a road-map of how each empirical study builds on the findings of the previous study is included in Figure 1.1.

Chapter 2 – Literature Review presents an overview of the literature of feedback, theories of motivation, and motivation in software engineering. Four theories of motivation that identify feedback as a factor are presented, and their interpretation and inclusion of feedback in each theory is critiqued. The 'state of play' in the understanding of motivation in software engineering is discussed, and the quality and reliability of prior research is investigated and presented. These findings show that previous research may not have sufficiently included theories of motivation when investigating motivation in software engineering. Characteristics of feedback as found in clinical education are identified, motivation is defined, and the relationship between feedback, motivation, and theories of motivation is established.

Chapter 3 – Feedback Scoping Study presents the findings of the preliminary study which aimed to identify any feedback characteristics reported by software engineers and compare them to characteristics found in another discipline. During this study experienced software engineers were interviewed and asked to discuss feedback. This study identified that feedback is described by software engineers as containing several characteristics, comparable to those identified in clinical education.

Chapter 4 – Research Design for the Feedback in One Software engineering Environment presents the research design for the first major study, the Feedback in One Software Engineering Environment Study (chapter 5). This research design builds on the findings from the Feedback Scoping Study (chapter 3), which found that software engineers discussed feedback characteristics, and reported that feedback can affect them. The foundations of the research design are presented, and the research methods used are identified and described. Ethical considerations are addressed, and the complete picture of the combination of multiple research methods to investigate a phenomenon from multiple angles is presented.

Chapter 5 – Feedback in One Software engineering Environment Study presents the findings from the study described in the research design (chapter 4). This study builds on the findings of the Feedback Scoping Study (chapter 3) and by identifying different forms of feedback and their characteristic values in software engineering environments. This study investigated the extended impact feedback can have on software engineers by exploring how software engineers feel at the end of their working day. Feedback is investigated with software engineers using multiple research methods. This study identified that a range of feedback commonly occurs in software engineering environments, and that the impact of received feedback is different for positive or negative feedback. The findings of this study raise questions that require investigation in additional studies.

Chapter 6 – The Short-Term Impact of Received Feedback presents a study that builds on the findings of the Feedback in One Software Engineering Environment Study (chapter 5), which identified that feedback is received by software engineers, that received feedback can affect software engineers' attitude, behaviour, job satisfaction, productivity, feelings and motivation, and that feedback can affect how software engineers' feel at the end of their working day. This study builds on these earlier findings by investigating the impact of the characteristics of received feedback, and identifying whether the values of feedback characteristics can change the impact of received feedback. The findings of this study highlight that the source and medium feedback characteristics can affect the impact of received feedback.

Chapter 7 – The Effect of 'Source' and 'Medium' Feedback Characteristics presents a study that investigated in more depth the impact of the source and medium feedback characteristics. This study builds on the findings of the Feedback Scoping Study (chapter 3), the Feedback in One Software engineering Environment Study (chapter 5), and the Short-Term Impact of Received Feedback Study (chapter 6). These studies identified that feedback is discussed by software engineers as having characteristics similar to those identified in clinical education, that feedback can affect software engineers' attitudes, behaviours, productivity, job satisfaction, and motivation, and that the different values of feedback characteristics, specifically the source, medium, and content polarity, can alter the effect of received feedback. This study investigates the values and impact of feedback received from a range of different sources, and investigates the impact of receiving feedback through different media. This study found that what software engineers report as being the most valuable feedback, is not the same as feedback that has the most impact.

Chapter 8 – A Model of Feedback in Software Engineering presents a combination of the findings from the studies presented in chapters 3, 4, 5 and 6, and relevant literature to identify a model capturing how feedback is received by software engineers. The individual perceptions, individual characteristics, decisions, and impact of received feedback all combine to produce a model of feedback in software engineering.

Chapter 9 – Discussion encapsulates the findings from all empirical data collection presented in chapters 3, 5, 6 and 7. The importance of the medium and source feedback characteristics is identified, and the impact that the different values of feedback characteristics have is identified. The findings are then compared to theories of motivation and previous literature, and the limitations of the research are discussed.

Chapter 10 – Conclusion presents the conclusion of the thesis. A final summary of the research is provided, the contributions from the research identified, and possible future avenues of research are discussed.

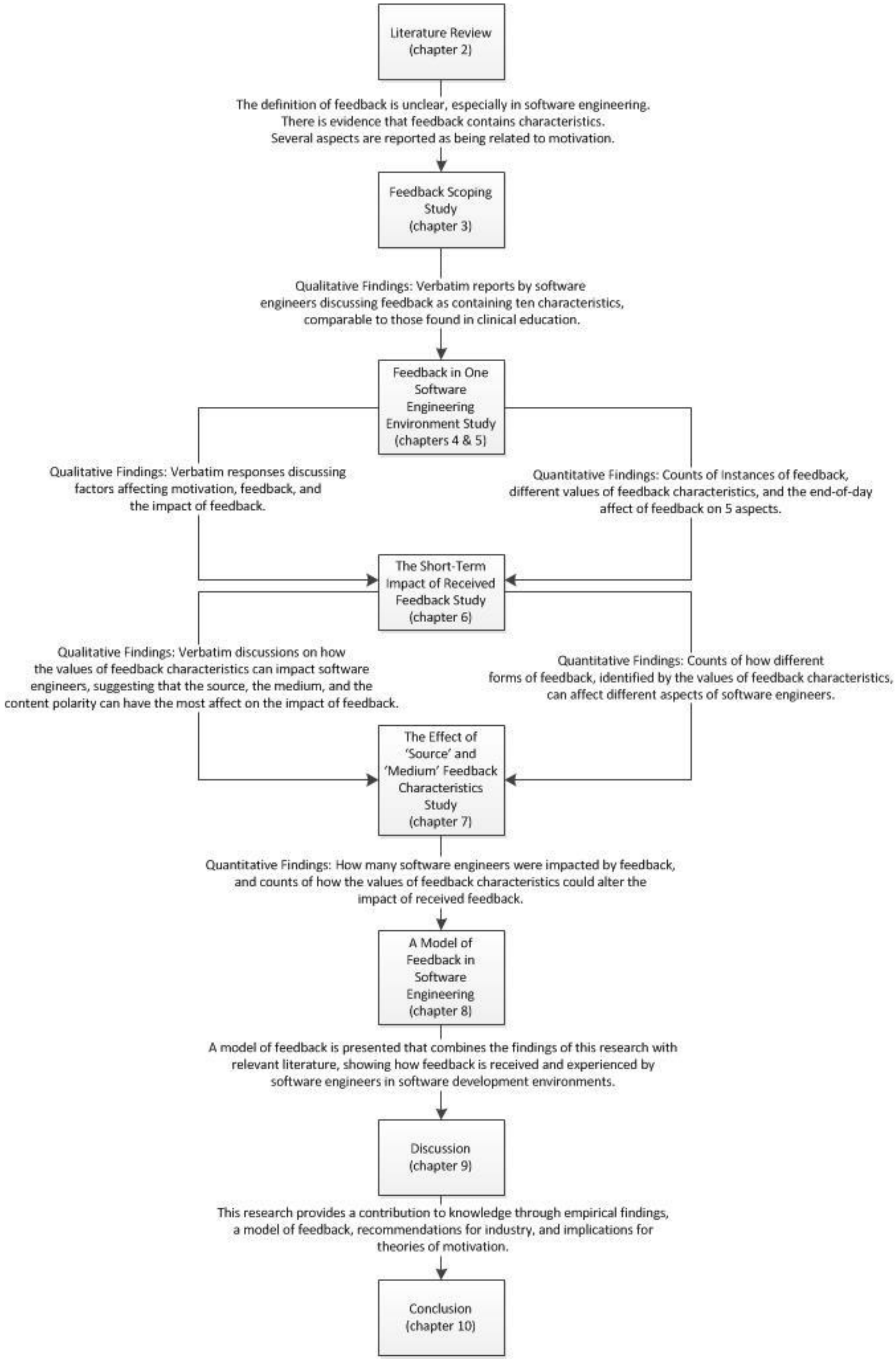

Literature Review (chapter 2)
The definition of feedback is unclear, especially in software engineering.
There is evidence that feedback contains characteristics.
Several aspects are reported as being related to motivation.
Feedback Scoping Study (chapter 3)
Qualitative Findings: Verbatim reports by software engineers discussing feedback as containing ten characteristics, comparable to those found in clinical education.
Feedback in One Software Engineering Environment Study (chapters 4 & 5)
Qualitative Findings: Verbatim responses discussing factors affecting motivation, feedback, and the impact of feedback.
Quantitative Findings: Counts of Instances of feedback, different values of feedback characteristics, and the end-of-day affect of feedback on 5 aspects.
The Short-Term Impact of Received Feedback Study (chapter 6)
Qualitative Findings: Verbatim discussions on how the values of feedback characteristics can impact software engineers, suggesting that the source, the medium, and the content polarity can have the most affect on the impact of feedback.
Quantitative Findings: Counts of how different forms of feedback, identified by the values of feedback characteristics, can affect different aspects of software engineers.
The Effect of 'Source' and 'Medium' Feedback Characteristics Study (chapter 7)
Quantitative Findings: How many software engineers were impacted by feedback, and counts of how the values of feedback characteristics could alter the impact of received feedback.
A Model of Feedback in Software Engineering (chapter 8)
A model of feedback is presented that combines the findings of this research with relevant literature, showing how feedback is received and experienced by software engineers in software development environments.
Discussion (chapter 9)
This research provides a contribution to knowledge through empirical findings, a model of feedback, recommendations for industry, and implications for theories of motivation.
Conclusion (chapter 10)

FIGURE 1.1 - ROAD-MAP OF THE RESEARCH PRESENTED IN THIS THESIS

CHAPTER 2

LITERATURE REVIEW

The need for a better understanding of motivation and the role it plays in software development environments has been recognised for many years (Couger and Zawacki, 1980). However, previous research has not investigated how feedback affects software engineers, and theories of motivation do not consider the effect of different forms of feedback.

This chapter reviews the literature to identify the current understanding of feedback and its role affecting the motivation of software engineers. The definition of motivation is investigated, theories of motivation that include feedback as a factor are identified, and previous research investigating motivation in software engineering environments is reviewed. The story emerging from the literature is one of uncertainty surrounding the definition of feedback, with theories of motivation presenting feedback as a single entity, and findings in clinical education arguing that feedback is the combination of multiple characteristics. Feedback is identified as a factor affecting the motivation of software engineers, but the affect received feedback has, and the impact of the values of different feedback characteristics, is unclear.

The following sections present the definition of motivation and theories of motivation (2.1), identify research investigating motivation in software engineering (2.2), identify and compare definitions of feedback (2.3), present findings of research investigating feedback (2.4), and finally provide a summary of the reviewed literature (2.5).

2.1 MOTIVATION

Motivation is a term that has been in use for over a century (James, 1884, reported in Berridge and Winkielman, 2003), but the definition of what motivation is and what constitutes 'being motivated' is disputed within and across disciplines. How motivation is viewed varies depending on the perspective being taken; to an individual it may be the drive that helps him/her continue to do his/her job, but to an employer it might be considered as a factor that can be influenced to improve the performance of an employee. Different disciplines and researchers have presented definitions of motivation, with a focus on the distinction between intrinsic and extrinsic motivation.

The literature agrees that there is a distinction between two types of motivation, identified as intrinsic and extrinsic. Herzberg argues the importance of intrinsic factors:

> *"Motivation is a function of growth from getting intrinsic rewards out of interesting and challenging work" (Herzberg, 1987: 14).*

Herzberg believes that true motivation is intrinsic, and not extrinsic. Ryan and Deci define *intrinsic motivation* as:

> *"the doing of an activity for its inherent satisfactions rather than for some separable consequence" (Ryan and Deci, 2000: 56).*

Ryan and Deci continue to define *extrinsic motivation*:

> *"Extrinsic motivation is a construct that pertains whenever an activity is done in order to attain some separable outcome" (ibid: 60).*

This is different from Herzberg, who doesn't believe that extrinsic motivation is actually motivation, and he defines this as *movement*:

> *"Movement is a function of fear of punishment or failure to get extrinsic rewards." (Herzberg, 1987: 14)*

Baron defines motivation as something which is an internal process directing behaviour:

> *"the internal processes that activate, guide, and maintain behaviour (especially goal-directed behaviour)" (Baron, 1991: 1).*

Robertson and Smith have a similar approach to Baron's:

> *"Motivation is a psychological concept related to the strength and direction of human behaviour." (Robertson and Smith, 1985: 2)*

Baron and Robertson and Smith describe the definition of motivation using a human-centric approach, suggesting that motivation is a psychological state and includes an internal set of processes. Herzberg and Ryan and Deci focus on motivation using a managerial perspective, and propose that motivation is a drive to attain internal satisfaction or other intrinsic rewards from the work that people do. These different definitions of motivation illustrate the lack of a consensus on what motivation is and how motivation is defined.

Theories of motivation exist that attempt to identify and model how motivation is affected. These theories look at motivation at different points in time and at different levels of granularity. Broadly, the theories can be classified as either focusing on motivation "at a single point in time" (Couger and Zawacki, 1980: 76): content theory, or focusing on motivation "as a sequence or process of related activities" (Hall *et al.*, 2009: 4): process theory. The following section discusses the theories of motivation relevant to this research.

2.1.1 THEORIES OF MOTIVATION

Most of the theories of motivation (Table 2.1) that are widely accepted and supported were developed in the 1950s and 1960s. Some of the theories are adaptations or evolutions of previous theories, for example, Alderfer's Existence-Relatedness-Growth theory builds on Maslow's Hierarchy of Needs theory.

Feedback is included as a factor affecting motivation in four theories of motivation listed in Table 2.1. Two of the theories of motivation that include feedback as a factor are 'process theories', and the other two theories of motivation that include feedback as a factor are 'content theories'; feedback is identified as a factor that can affect motivation during “a sequence of process related activities” (Hall *et al.*, 2009: 4) and “at a single point in time” (Couger and Zawacki, 1980: 76). In the following sections, each of the four theories is presented, and their inclusion of feedback and its role in the theory is identified: Achievement theory, Goal Setting theory, Hygiene theory, and Job Characteristics theory.

TABLE 2.1 – THEORIES OF MOTIVATION[1]

Name	Author
Achievement Theory	McClelland 1961
Activation Theory	Berlyne 1967
Cognitive Evaluation Theory	Deci 1975
Control Theory	Wiener 1948, Miller, Galanter and Pribram 1960
Drive Theory	Hull 1943
Equity Theory	Adams 1963
Existence-Relatedness-Growth Theory	Alderfer 1969
Expectancy Theory	Vroom 1964
Goal Setting Theory	Locke 1968
Hierarchy of Needs Theory	Maslow 1954
Hygiene Theory	Herzberg 1959
Internal-External Control Theory	Rotter 1966
Job Characteristics Theory	Hackman & Oldham 1976
Stimulus Response Theory	Skinner 1976
Theory X and Y	McGregor 1960

ACHIEVEMENT THEORY

Achievement Theory (McClelland, 1961) proposes that there are three key motivational needs that most people strive to satisfy: achievement, power, and affiliation.

McClelland defined an achievement-oriented activity as one in which the individual feels responsible for the outcome, has expectations of unambiguous feedback, and in which there is risk or uncertainty of the outcome. Individuals feel

[1] Underlined theories of motivation include feedback as a factor affecting motivation.

motivated to pursue these tasks if the tasks are ones the individual wants to do, the outcome of the activity is seen as achievable, and the outcome is of value to the individual.

Atkinson (1964) discussing McClelland’s achievement theory describes the need for achievement as the:

> *“Capacity to experience pride in accomplishment” (Atkinson, 1964: 241).*

Accomplishment could be described as the link between expectation and the incentive/value of the outcome.

McClelland argues that every individual has a need for power: the need to have impact on other people, either through *personalised power* or *socialised power*.

- *Personalised power* is a type of power where individuals view situations as competitive, and where they can use their power to win the competitive situation. People with high personalised power are usually low on inhibition, and view relationships with other people as personal. McClelland claims that people with high personalised power are highly competitive and want to win or dominate someone else.

- *Socialised power* is a type of power where individuals do not view situations as competitive, and where they use their abilities for the good of the group. They view relationships with others as impersonal, and carefully plan conflict with other people in advance because they recognise that for every win there is a loss.

McClelland identified that an individual’s need for power can be shown in three different ways:

– Through strong actions, such as assault and aggression, giving help or assistance, controlling others, influencing or persuading others, or trying to impress someone.

– Through actions that produce strong emotions in others.

– Through actions that would enhance or preserve a person’s reputation.

McClelland characterised people with a high need for power as people who accumulate prestige possessions, play competitive one-to-one sports, like to belong to organisations and to take positions of power within them, and satisfy their need for power through thought and feeling.

McClelland argued that people have a need for affiliation; this is an individual's need to feel and be affiliated with other individuals, such as creating and maintaining positive relationships. Conflict is often avoided due to fear of rejection, and people with high nAff work hard to be accepted.

Feedback is included in achievement theory as one of the three key factors defining an achievement-oriented activity. An individual will expect to receive unambiguous feedback on the outcome of their actions.

GOAL SETTING THEORY

Goal Setting Theory (Locke, 1968) proposes that people are motivated to successfully complete challenging goals. From his research, Locke concluded that:

- Difficult goals produce a higher level of performance than easy goals.
- Goals that are specific produce higher levels of performance than vague goals.
- Individual behavioural intentions influence the choices that people make.

Locke stated that the goals which result in the highest levels of performance are difficult specific goals that are accepted by the participant. Locke identified six incentives for goal-setting:

1. Money
2. Knowledge of score
3. Time limits
4. Participation
5. Competition
6. Praise or reproof

The following sections present Locke's discussion on the effect of each of the identified incentives:

• MONEY

From his research, Locke found that money was able to assist in encouraging commitment to a task to which the person may not otherwise have been committed. However, Locke found that money in itself did not motivate above a certain level of income, even in situations where there was a direct link between earnings and output.

- KNOWLEDGE OF SCORE

Locke found that knowledge of score, provided through feedback, assisted motivation only when it was considered to be valid and where it could be used to set future goals. Knowledge of score received from someone who is not considered to be in a position to give feedback would not be regarded as valid feedback and would not affect motivation. Knowledge of score also had to be specific enough to be useful in the setting of future goals; otherwise it would not affect motivation.

Locke warned that the level of the standard of knowledge of score feedback given can influence the level of goals that are motivating. For example, if positive feedback is received for a low-level standard (e.g., reading more papers than the minimum level), the goals that the individual will likely be motivated to achieve will be lower than in a situation where feedback was given only once a higher-level standard was met (e.g., reading more papers this month than the month before).

- TIME LIMITS

Challenging time limits were found to increase the motivation to achieve a goal. But if a time limit were set that was unrealistic for the task, the task would likely take longer to complete than if a realistic but challenging time limit were set.

- PARTICIPATION

An individual's participation in the setting of goals was found to increase their motivation to achieve the goal, but it alone was not enough and required other factors for the task to be motivating. Locke reports "while subordinate participation in the goal-setting process had some effect on improved performance, a much more powerful influence was whether goals were set at all." (ibid: 179) It would appear that participation is good at increasing motivation and commitment to a goal rather than being the initial impetus to complete the goal.

- COMPETITION

Locke identified competition as a key motivator: when the performance of another individual or group is used to set a standard and to assess the success or failure of a goal. Locke believes that competition is a key factor in motivating people to continue to strive to meet higher levels of performance, especially when the high level is set by someone other than themselves. Locke argued that "If mile runners only ran against themselves or against a stopwatch, the 4-minute mile might never have been broken" (ibid: 180). Locke also believes competition is linked to innovation, with the aim of eventually getting better performance through better procedures, better products etc.

- PRAISE OR REPROOF

From his research, Locke identified a varying effect of praise or reproof on motivation. He found that praise was effective in improving overall performance, and that reproof appeared to improve the performance if feedback was provided in relation to a standard.

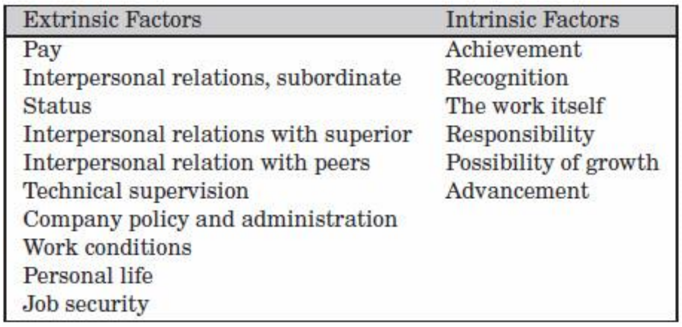
SUMMARY

Locke argued that six incentives have a varying level of influence on the setting of future goals. He reported that the most direct way of influencing goals is through time limits. Money, participation, praise and reproof are considered to be an indirect way of influencing goals, and knowledge of score and competition are seen as ways of suggesting a standard for future goals. Feedback is identified by goal setting theory and is used to provide an individual with the outcome of their actions (knowledge of score), and specific forms of feedback, praise and reproof, are identified as factors impacting performance.

HYGIENE THEORY

Hygiene Theory (Herzberg, 1959) proposes that there is a difference between what provides job satisfaction (intrinsic factors) and what avoids job dissatisfaction (extrinsic factors). Herzberg argues that job satisfaction and job dissatisfaction are separate, and that they are *not* the opposite of each other. Herzberg states that the opposite of satisfaction is no satisfaction, and the opposite of dissatisfaction is no dissatisfaction.

The extrinsic and intrinsic factors identified by Herzberg are listed in Table 2.1.



Extrinsic Factors	Intrinsic Factors
Pay	Achievement
Interpersonal relations, subordinate	Recognition
Status	The work itself
Interpersonal relations with superior	Responsibility
Interpersonal relation with peers	Possibility of growth
Technical supervision	Advancement
Company policy and administration	
Work conditions	
Personal life	
Job security	

FIGURE 2.1 – HYGIENE THEORY EXTRINSIC AND INTRINSIC FACTORS

Intrinsic factors (motivators) are aspects of a job which are satisfying, and efforts to improve them will improve motivation. Extrinsic factors (movement) are aspects of the job which avoid job dissatisfaction, and efforts to improve them will not improve motivation, but will avoid job dissatisfaction.

Salary was the only factor to appear on both the job satisfaction and job dissatisfaction lists, but Herzberg believed that the reason salary was mentioned as a job satisfaction factor was more to do with its links to recognition rather than salary itself, and subsequently he decided it was a movement factor rather than a motivator (Herzberg, 1959).

Overall, Herzberg's empirical findings suggested that people are more satisfied by aspects of the job than by the environment with in which they work, and that personal life appeared to have no real effect on job satisfaction or job dissatisfaction.

Herzberg identified that job satisfaction was linked to turnover of staff, attitude towards the company, mental health and interpersonal relationships through the positive or negative impact on job attitude:

- Turnover of staff: Dissatisfied people are more likely to quit, but Herzberg also discussed 'psychological quitting', when individuals will not leave the company, but will lower their effort on the job, perhaps only putting in the effort required to keep them employed.
- Attitude towards the company: Herzberg found a close link between job satisfaction and a positive attitude toward the company.
- Mental health: Herzberg reported that people link poor mental health and job dissatisfaction, but not good health and job satisfaction.
- Interpersonal relationships: Herzberg identified a minimal relationship between job satisfaction or job dissatisfaction and interpersonal relationships, and argued that this was because "the degree to which a person lets his feelings about his job spill over into the conduct of his interpersonal relationships is more a function of psychological dynamics as an individual than of anything else" (ibid: 93).

Herzberg concluded Hygiene Theory by claiming that if job satisfaction leads to greater productivity, motivators should lead to an improvement in performance, and he suggested that the work itself, responsibility and advancement bring about the most long-term improvements to job satisfaction. Herzberg argued that jobs could be improved by increasing the opportunities available to the employee to achieve goals that are meaningfully related to the job, even if they are not interesting. Herzberg proposed that jobs should be “set up in such a way that, interest or no, the individual who carries them out can find that their operations lead to increased motivation” (ibid: 134).

Feedback is identified in Hygiene Theory as the most common way recognition is received. Recognition is reported as the second strongest motivator, and can be received by almost anyone: “supervisor, some other individual in management, management as an impersonal force, a client, a peer, a professional colleague, or the general public” (ibid: 45). Recognition is identified as when “some act of notice, praise, or blame” (ibid: 45).

JOB CHARACTERISTICS THEORY

Job Characteristics Theory (Hackman and Oldham, 1976) proposes that there is a relationship between five core job characteristics (Skill Variety, Task Identity, Task Significance, Autonomy, Feedback) to three psychological states (Meaningfulness of Work, Responsibility of Outcomes, Knowledge of Actual Results) which in turn lead to four personal and work outcomes (High internal work motivation, High quality work, High satisfaction with work, Low absenteeism and turnover). Hackman and Oldham found that all of this is moderated to some extent by each individual’s growth need strength. The job characteristics model is shown below in Table 2.2.

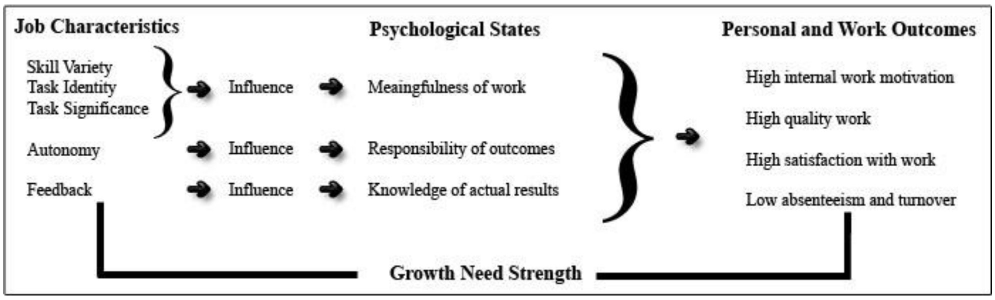


FIGURE 2.2 – JOB CHARACTERISTICS THEORY MODEL OF MOTIVATION

PSYCHOLOGICAL STATES

The core of the Job Characteristics Model is the three psychological states. Hackman and Oldham argue that individuals experience “positive affect” when they *know* (Knowledge of Actual Results) that *they* (Responsibility of Outcomes) have performed well on something they *care about* (Meaningfulness of Work).

Hackman and Oldham argue that this positive affect is reinforcing for the individual, and provides an incentive for the individual to continue to try and perform well in the future. When the individual fails to perform well, they will not experience these internal rewards, which may lead to them trying harder to regain the internal rewards in the future through better performance. This results in a "self-perpetuating cycle of positive work motivation powered by self-generated rewards that is predicted to continue until one or more of the three psychological states is no longer present, or until the individual no longer values the internal rewards that derive from good performance." (ibid: 256)

Hackman and Oldham define the three psychological states as:

> "*Experienced Meaningfulness of the Work: The degree to which the individual experiences the job as one which is generally meaningful, valuable and worthwhile.*
> *Experience Responsibility for Work Outcomes: The degree to which individual feels personally accountable and responsible for the results of the work he or she does.*
> *Knowledge of Results: The degree to which the individual knows and understands on a continuous basis, how effectively he or she is performing the job." (ibid: 256-257)*

The following sections present and discuss the job characteristics that influence the three psychological states.

The three job characteristics (skill variety, task identity, task significance) that influence the psychological *meaningfulness of a job* are defined by Hackman and Oldham as:

> "*Skill Variety: The degree to which a job requires a variety of different activities in carry out the work, which involve the use of a number of different skills and talents of the person." (ibid: 257)*

Tasks are almost always experienced as meaningful when they require the individual to engage in different aspects that challenge or stretch their skills and abilities. Hackman and Oldham believe that this is still true on jobs that draw upon multiple skills of individuals, even if the jobs are not that significant or important.

> *"Task Identity: The degree to which the job requires completion of a "whole" identifiable piece of work; that is, doing a job from beginning to end with a visible outcome." (ibid: 257)*

The individual should find jobs that require a complete product or a complete unit of service more meaningful than if they were only completing a small part of the product or unit of service.

> *"Task Significance: The degree to which the job has substantial impact on the lives or work of other people, whether in the immediate organisation or in the external environment." (ibid: 257)*

The meaningfulness of the work is usually enhanced when the individual understands the impact that their work may have on the well-being of other people.

The job characteristic (autonomy) which influences the psychological *experienced responsibility* is defined by Hackman and Oldham as:

> "*Autonomy: The degree to which the job provides substantial freedom, independence, and discretion to the individual in scheduling the work and in determining the procedures to be used in carrying it out." (ibid: 258)*

When the outcome of the task/project/system depends to some degree on the individual's own efforts, initiatives and decisions rather than on the quality of instructions from the boss or on a manual of job procedures, it is likely that the individual will feel strong personal responsibility for the outcome of the task/project/system.

The job characteristic (feedback) which influences the psychological *knowledge of results* is defined by Hackman and Oldham as:

> *"Feedback: The degree to which carrying out the work activities required by the job results in the individual obtaining direct and clear information about the effectiveness of his or her performance." (ibid: 258)*

Hackman and Oldham don't discuss feedback or the influence it has any further; it is simply a tool to communicate knowledge of the results of an action to an individual, and it is not discussed as having any other impact.

PERSONAL AND WORK OUTCOMES

Hackman and Oldham identified that the presence of the three psychological states leads to the following personal and work outcomes:

- High internal work motivation.
- High quality work.
- High satisfaction with work.
- Low absenteeism and turnover.

They expect these outcomes to be more positive with jobs scoring higher in the motivating potential score (see below) than with jobs scoring lower. The model does not address the priorities of these outcomes or consider any ordering.

MOTIVATING POTENTIAL SCORE

Hackman and Oldham proposed that it is possible to measure the potential for a job to be motivating, and argued that the potential for a job to promote internal work motivation for individuals should be at its highest when all of the following criteria are met:

a) The job is high in at least one (and hopefully more) of the three job characteristics that lead to experienced meaningfulness (i.e., skill variety, task identity, and task significance).

b) The job is high on autonomy.

c) The job is high on feedback.

The Motivating Potential Score measures the degree to which the above criteria are met:

TABLE 2.2 – JOB CHARACTERISTICS THEORY MOTIVATION POTENTIAL SCORE

$$\text{Motivating Potential Score (MPS)} = \left| \frac{\text{Skill Variety} + \text{Task Identity} + \text{Task Significance}}{3} \right| \text{X} \ \text{Autonomy} \ \text{X} \ \text{Feedback}$$

A zero score on either autonomy or feedback will reduce the overall score to near zero, whereas a near zero score on one of the three job characteristics contributing to experienced meaningfulness (i.e., skill variety, task identity, and task significance) will not by itself lead to an overall near-zero score.

GROWTH NEED STRENGTH

Growth Need Strength is Hackman and Oldham’s concept to address the differences present from individual to individual and to identify in what ways they influence how each individual reacts to their work. Hackman and Oldham suggest that people who have a high need for personal growth and development will respond positively to jobs that are high in motivation potential, and that people with a low growth need will respond less positively to the same job.

Hackman and Oldham believe that the effect of individual growth need strength may be present in two areas (shown in Figure 2.2):

a) The link between the objective job characteristics and the psychological states.
b) The link between the psychological states and the outcome variables.

It is suggested that if the effect of individual growth need strength is found at (a), then high-growth-need individuals are more likely or better able to *experience* the psychological states than low-growth-need strength individuals, when their job has a good motivation potential score. If the effect of individual growth need strength is found at (b), then it is suggested that nearly everyone may experience the psychological states when their job has a good motivation potential score, but that the individuals with high growth need will respond more positively to the experience. It is also suggested that it is possible that growth need strength can have an effect at both locations (a) and (b) in the model. Feedback is included in Job Characteristics Theory as one of 5 key factors required for a job to be motivating. Feedback provides the individuals with the knowledge of the effectiveness of their performance, and influences the psychological state of ‘knowledge of actual results’.

2.1.2 THEORIES OF MOTIVATION SUMMARY: FEEDBACK

The four theories presented in the previous sections identify feedback as a factor affecting motivation. The validity of the feedback as perceived by the receiver and the difference between performance and affiliation feedback are the only extractable characteristics of feedback discussed by any of the four theories.

The four theories of motivation provide similar definitions of feedback, and all use feedback to transfer information. Feedback is used in all of the theories as a method for providing information about an individual’s performance. All four of the theories of motivation focus on feedback as providing information, and do not identify other effects of receiving feedback, nor do they consider the effect of different feedback characteristics.

Job Characteristics Theory uses feedback as a method of providing individuals with the knowledge of the outcome of their performance. This directly influences the psychological feeling “knowledge of the results of their work” (Hackman and Oldham, 1976: 257), which is an essential psychological state required for any job to be motivating. Job Characteristics Theory does not define any characteristics of feedback, and does not discuss the impact of different forms of feedback. Job Characteristics Theory implies that the only important characteristic is the information transferred to the recipient by feedback.

In Goal Setting Theory feedback is used to provide individuals with the information of how well they have completed a goal. Goal Setting Theory argues that knowledge of score (feedback) is used to influence future goals and to suggest standards of performance. The validity of feedback is discussed; it is claimed that feedback must be given by a person who is perceived as being in a position to provide valid feedback. Goal Setting Theory identifies two specific forms of feedback, praise and reproof, and suggests that these forms of feedback can affect future goal setting.

Hygiene Theory identifies recognition as an important factor influencing motivation, and claims that recognition is most often received through “verbal” feedback (Herzberg, 1959: 45). Recognition includes feedback both to provide praise for a job well done and to provide knowledge of the results of one’s actions. Neither the importance of received feedback nor the different characteristics that any received feedback may have is discussed.

Theory of Needs identifies unambiguous feedback as one of the three key factors required for any activity to be an achievement-oriented activity. The knowledge of the results of one’s actions in achievement-oriented activities is attained through feedback. Theory of Needs does not identify different characteristics of feedback or discuss any possible importance that these characteristics may have.

In summary, four theories of motivation were presented that identified feedback as a factor in motivation, and all four theories suggest that feedback is used to provide the recipient with either the knowledge of the outcome of their actions or with recognition for something he/she has done. While some theories of motivation identify specific forms of feedback (praise and reproof as in Goal Setting Theory), and unremarked characteristics can be identified (feedback validity, as seen in Goal Setting Theory), none of the theories discusses the importance of feedback other than to provide an individual with information, nor do they consider the impact of any characteristics of feedback.

2.2 MOTIVATION IN SOFTWARE ENGINEERING

Studies of motivation in software engineering have identified that motivation, and dealing with motivation, can affect several aspects of software engineers. DeMarco and Lister (1999) identified motivation as one of the most frequently-cited causes of an unsuccessful software project. Abdel-Hamid (1989) argued that the real cost of dealing with unmotivated software engineers could be as much as 60% of a project's budget. Motivation can impact the quality of software (Boehm, 1981), the overall success of the project (Hall *et al.*, 2008), and the productivity of software engineers (Procaccino *et al.*, 2005).

Motivation in software engineering has been investigated by academics since the 1970s. A large body of work exists from the 1970s and 1980s, but could be considered out of date due to evidence (Beecham *et al.*, 2008) supporting claims that software engineering motivators and the software engineering discipline itself has evolved during this time (Sharp and Hall, 2009). Recently, Amabile and Kramer (2011) identified the importance of making progress for professionals, with software engineers included in their study.

A systematic literature review of motivation in software engineering was conducted by Beecham *et al.* (2008) to answer five research questions:

> *"RQ1: What are the characteristics of Software Engineers?*
> *RQ2: What (de)motivates Software Engineers to be more (less) productive?*
> *RQ3: What are the external signs or outcomes of (de)motivated Software Engineers?*
> *RQ4: What aspects of Software Engineering (de)motivate Software Engineers?*
> *RQ5: What models of motivation exist in Software Engineering?" (ibid: 861)*

From online publication database searches with strict pre-set criteria Beecham *et al.* (2008) identified over 2000 papers published before March 2006, and by reading each papers abstract and title they found a sub-set of just over 500 papers. After reading in full the 500 papers, Beecham *et al.* identified 92 papers relevant for their systematic literature review. They identified 21 motivators and 15 de-motivators present in the literature, shown in Table 2.3 and Table 2.4.

TABLE 2.3 – MOTIVATORS IN SOFTWARE ENGINEERING (BEECHAM *ET AL.*, 2008: 868)

Motivators	**Frequency** **(# of studies out of 92)**
Rewards and incentives	14
Development needs addressed	11
Variety of work	14
Career path	15
Empowerment/responsibility	6
Good management	16
Sense of belonging/supportive relationships	14
Work/life balance	7
Working in a successful company	2
Employee participation/involvement/working with others	16
Feedback	10
Recognition	12
Equity	3
Trust/respect	4
Technically challenging work	11
Job security/stable environment	10
Identify with the task	20
Autonomy	9
Appropriate working conditions/environment/good equipment/tools/physical space/quiet	6
Making a contribution/task significance	6
Sufficient resources	2

TABLE 2.4 – DE-MOTIVATORS IN SOFTWARE ENGINEERING (BEECHAM *ET AL.*, 2008: 869)

De-Motivators	**Frequency** **(# of studies out of 92)**
Risk	1
Stress	5
Inequity	4
Interesting work going to other parties	1
Unfair reward system	2
Lack of promotion opportunities/stagnation/career plateau/boring work/ poor job-fit	5
Poor communication	5
Uncompetitive pay/poor pay/unpaid overtime	6
Unrealistic goals/phoney deadlines	4
Bad relationship with users and colleagues	4
Poor working environment	9
Poor management	7
Producing poor quality software	3
Poor cultural fit/stereotyping/role ambiguity	3
Lack of influence/not involved in decision making/no voice	2

França *et al.* (2011) extended this research by analysing 53 papers covering March 2006 to August 2010 using an approach heavily based on the Beecham *et al.* (2008) research. França *et al.* identified 8 additional motivators shown in Table 2.5, while also reporting that 2 of the original motivators from the Beecham *et al.* study were not present (appropriate working conditions, sufficient resources). França *et al.* identified one additional de-motivator from the reviewed literature (task complexity).

TABLE 2.5 – NEW MOTIVATORS IN SOFTWARE ENGINEERING (FRANÇA *ET AL.*, 2011: 159)

Motivators	**Frequency** **(# of studies out of 53)**
Team quality	4
Creativity/innovation	4
Fun	1
Professionalism	2
Having an ideology	1
Non-financial benefits	1
Penalty policies	1
Good relationship with users/customers	2

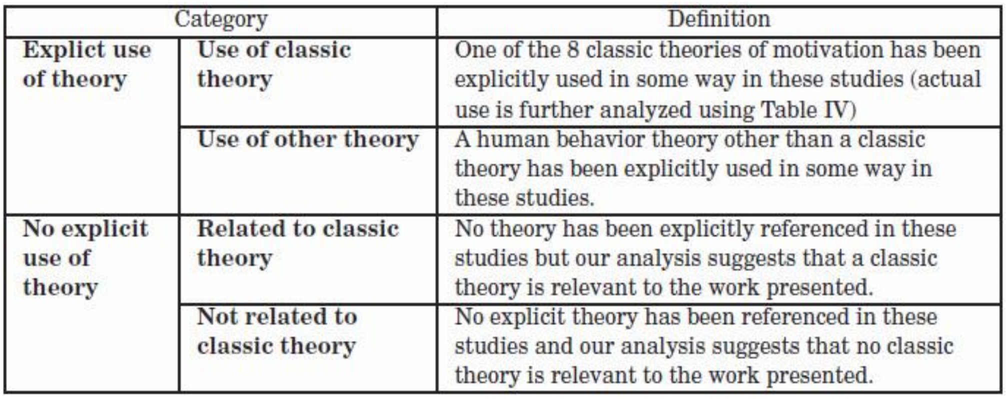
In both the 2008 and 2011 systematic literature reviews, there is a variation in the frequency in which each motivator and de-motivator is reported in the reviewed studies. This makes the effect on software engineers and the value to software engineers of each motivator and de-motivator unclear; however assessing value and effect of a factor was not the intention of either systematic literature review.

The addition of 8 motivators and 1 de-motivator supports claims that the motivation of software engineers has evolved since the majority of research was conducted (Sharp and Hall, 2009), and suggests that motivation will continue to change as the discipline evolves, necessitating further research.

The reliability and accuracy of the studies reviewed in the Beecham *et al.* systematic literature review is unclear. Over 80% of the reviewed studies collected their data through questionnaires (examples: Procaccino *et al.* (2005), Khalil *et al.* (1997), Couger and Ishikawa (1985)), and often remotely-administered questionnaires (examples: Hertel *et al.* (2003), Couger and Adelsberger (1988)).

2.2.1 THEORIES OF MOTIVATION IN SOFTWARE ENGINEERING

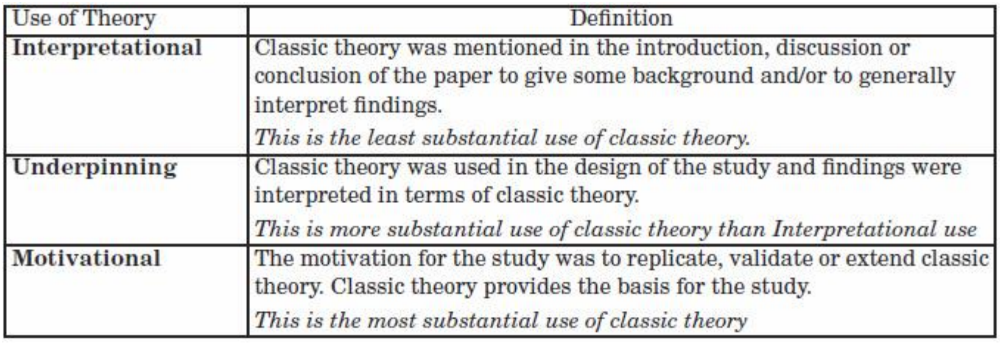
Hall *et al.* (2009) dispute the foundations of the investigations of the papers reviewed by Beecham *et al.* based on their use of motivation theory. Following their systematic literature review, the same group of researchers evaluated the use of theory in these studies. Their review identified the inadequate use of motivation theories by the majority of previous studies investigating motivation in software engineering. Hall *et al.* grouped the use of theory in the reviewed literature using the categories shown in Figure 2.3. Hall *et al.* listed what they described as “The Eight Classic Motivation Theories” (ibid: 4): Equity Theory, Stimulus Response Theory, Job Characteristics Theory, Goal Setting tTheory, Expectancy Theory, Need Theory – Maslow, Need Theory – McClelland, Motivation-Hygiene Theory. During this research, “Need theory – McClelland” is referred to as Achievement Theory, and “Motivation-Hygiene Theory” is referred to as Hygiene Theory. Note the reference in Figure 2.3 to Table IV is in this thesis a reference to Table 2.4.



Category		Definition
Explict use of theory	**Use of classic theory**	One of the 8 classic theories of motivation has been explicitly used in some way in these studies (actual use is further analyzed using Table IV)
	Use of other theory	A human behavior theory other than a classic theory has been explicitly used in some way in these studies.
No explicit use of theory	**Related to classic theory**	No theory has been explicitly referenced in these studies but our analysis suggests that a classic theory is relevant to the work presented.
	Not related to classic theory	No explicit theory has been referenced in these studies and our analysis suggests that no classic theory is relevant to the work presented.

FIGURE 2.3 – CATEGORISATION OF THEORY USE (HALL *ET AL.*, 2009: 10)

Hall *et al.* (2009) further define three types of explicit use of theory:



Use of Theory	Definition
Interpretational	Classic theory was mentioned in the introduction, discussion or conclusion of the paper to give some background and/or to generally interpret findings. *This is the least substantial use of classic theory.*
Underpinning	Classic theory was used in the design of the study and findings were interpreted in terms of classic theory. *This is more substantial use of classic theory than Interpretational use*
Motivational	The motivation for the study was to replicate, validate or extend classic theory. Classic theory provides the basis for the study. *This is the most substantial use of classic theory*

FIGURE 2.4 – CATEGORISATION OF EXPLICIT THEORY USE (HALL *ET AL.*, 2009: 11)

After reviewing the 92 identified papers, Hall *et al.* found that only 51 papers explicitly used at least one of the 8 classic theories of motivation. Further analysis of their results, as shown in Table 2.6 identifies that, for each of the four theories relevant to this research, the number of papers that were considered to have a 'motivational' use of each theory was low. It should be noted, each reviewed paper may include more than one theory of motivation.

TABLE 2.6 – EXPLICIT USE OF THEORY IN RELEVANT THEORIES OF MOTIVATION (BASED ON HALL *ET AL.*, 2009 FINDINGS)

Theory	Total	Interpretational	Underpinning	Motivational
Job Characteristics Theory	35	12	8	15
Goal Setting Theory	9	7	1	1
Achievement Theory	10	8	1	1
Hygiene Theory	21	8	6	7

Of the reviewed studies using any of the four theories of motivation relevant to this research, just 32% (24/75) of theory use was identified as "motivational" in the research.

The following sub-sections present and discuss the use of each of the relevant theories in the reviewed literature. This includes literature present in both the Beecham *et al.* (2008) and França *et al.* (2011) systematic literature reviews, as well as other relevant literature outside of the scope of the systematic literature reviews.

2.2.2 ACHIEVEMENT THEORY

Achievement Theory was used by Agarwal and Ferratt (2001) to investigate the failure of the section to meet the needs of IT workers. One of their suggested changes to practice was the inclusion of an individual performance measurement and the production of short-term and long-term organised goals. The production of goals would help provide IT workings with the sense of achievement that would be experienced by meeting them.

LeDuc (1980) reviewed the current state of play in software engineering, and argued that some programmers were being left unsatisfied and "cheated" due to a lack of intermediate goals:

> *"The programmer who works on a three-year development project without meaningful intermediate goals is being cheated out of accomplishment feedback." (ibid: 11)*

LeDuc also believed that, like some software engineers who had fewer short-term goals which provided them with fewer tasks to achieve, others were being neglected of any form of goal, long-term or short-term:

> *"The maintenance programmer whose duties consist of "solving whatever problem comes up today" is also being cheated, since day follows day without specific accomplishment." (ibid: 11)*

LeDuc continued to argue for what he felt was an urgent requirement to ensure that software engineers were provided with adequate goals and feedback for them to feel the inherent accomplishment from successfully meeting set goals. By providing software engineers with goals, software engineers could achieve, and could feel the sense of achievement.

Enns *et al.* (2006) discussed some stereotypes given to IT personnel, and suggested that the stereotypes do not go far enough towards identifying the needs of IT personnel, arguing that the motivations of IT professions *"cut across age and organisational tenure profiles" (ibid: 109)*. However, their findings and the stereotypes they discussed did align to suggest that software engineers were motivated by challenging goals and by clear feedback.

2.2.3 GOAL SETTING THEORY

Burn *et al.* (1992) used Goal Setting Theory to look specifically at the computing profession in Hong Kong, where also using the Job Characteristics Model they identified a significant difference between the need for growth as reported by employees, and the evaluated motivational potential score of the job. One key issue identified was the manager's inability to include employee participation when setting goals aimed at improving performance, which Goal Setting Theory argues is an important factor of any goal.

Rasch and Tosi (1992) investigated factors affecting software developers' performance. Their respondents to a remote questionnaire were software developers who were each involved in all stages of development from the beginning of the project to the end. They were able to identify, from their research, that goal difficulty had a negative relationship with performance, but a positive relationship with effort, and that goal clarity had a relatively small effect on performance.

Couger and Adelsberger (1988) compared the motivation of Japanese computer personnel to United States personnel using remote questionnaires. Their research suggested that there is a greater need for goal-specific feedback to be given by management to their employees in Japan, while this was not an issue for the United States computing personnel.

2.2.4 HYGIENE THEORY

Hygiene Theory was included in a study by McLean *et al.* (1996). Their research specifically addressed the need to provide more frequent feedback to IS professionals, and their conclusion stated that:

> *"The challenge of I/S managers was appropriately stated by Herzberg nearly forty years ago: 'If you want people motivated to do a good job, give them a good job to do.'" (ibid: 298)*

Their research supports Herzberg and his advice suggesting the re-design of jobs to make them more motivational and attractive (Herzberg, 1959).

Mak and Sockel (2001) completed a factor analysis of IS employee motivation and retention and identified perception of advancement as key to positively influencing motivation. They also identified loyalty, burnout and turnover as indicators for retention, or lack thereof.

Igbaria *et al.* (1995) completed research with 112 IS employees in South Africa, where they noted that while managers scored high on managerial competence and job security, they scored lower on lifestyle than professional positions, suggesting a difference in the motivational factors of managers and software engineers.

2.2.5 JOB CHARACTERISTICS THEORY

Job Characteristics Theory was used explicitly in the most substantial way (classified by Hall *et al.* (2009) as a 'motivational' use of theory) in more papers than any of the three other relevant theories. Couger and Zawacki (1980) modified the JCT model to address motivation in software engineering based on the findings of their research, as shown in Figure 2.5. They argued that the identification and inclusion of goals was crucial to software engineering, which relates their modified model of the Job Characteristics Model to Locke's Goal Setting Theory. Couger and Zawacki (1980) suggested that feedback required further definition in order to be accurate for software engineers, identifying 3 specific forms of feedback (Figure 2.5). Couger and Zawacki (1980) suggested the inclusion of social need strength (SNS). SNS is a measurement of an individual’s wish to interact with others on and off the job.

Studies using Job Characteristics Theory include Khalil *et al.* (1997) who looked specifically at the motivations of Egyptian IS employees. They completed a survey with 107 various IS personnel from 14 different organisations, and concluded that while the IS field in Egypt attracts individuals with a high need for growth, the jobs currently do not provide a high motivation potential to match the needs of the employees.

Procaccino *et al.* (2005) developed a 50-question questionnaire using 5-point Likert scales based on motivation literature, and they gathered responses from 66 software professionals. From their questionnaire and subsequent data analysis, Procaccino *et al.* were able to conclude that software engineers *“consider software projects successful if they provide intrinsic, internally motivating work to develop software systems that both meet customer/user needs and are easy to use.” (ibid: 200)*

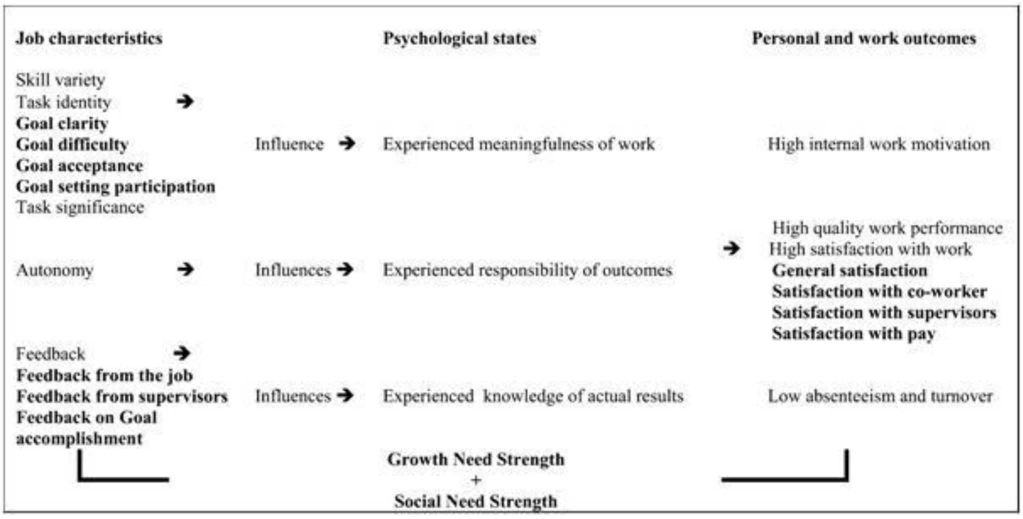

Job characteristics
Psychological states
Personal and work outcomes
Skill variety
Task identity
Goal clarity
Goal difficulty
Goal acceptance
Goal setting participation
Task significance
Influence
Experienced meaningfulness of work
High internal work motivation
High quality work performance
High satisfaction with work
General satisfaction
Satisfaction with co-worker
Satisfaction with supervisors
Satisfaction with pay
Autonomy
Influences
Experienced responsibility of outcomes
Feedback
Feedback from the job
Feedback from supervisors
Feedback on Goal accomplishment
Influences
Experienced knowledge of actual results
Low absenteeism and turnover
Growth Need Strength
+
Social Need Strength

FIGURE 2.5 – MODIFIED JCT MODEL OF MOTIVATION (COUGER AND ZAWACKI, 1980)

2.2.6 THEORIES OF MOTIVATION RESEARCH SUMMARY

While there is a significant body of research on motivation in software engineering, most of the research was completed over a decade ago, and could be considered out-dated in a changing software development industry.

Previous research often relied on using questionnaires, specifically remote questionnaires, to investigate motivation in software engineering. While questionnaires are useful for collecting a broad range of data, they lack the flexibility, adaptability and expansibility required when investigating a *state of mind* such as motivation.

The reviewed literature often did not make sufficient use of theories of motivation when investigating the motivation of software engineers. The use of theories of motivation in previous research investigating motivation in software engineering suggests that the studies had insufficient grounding in relevant literature, questioning the robustness of their findings (Hall *et al.*, 2009).

2.3 FEEDBACK

Feedback is often used in the workplace as part of the employee management process. Praise, criticism, and recognition are all forms of feedback, when someone is reacting or responding to something a colleague has done. Feedback can be defined (dictionary.com, 2012) as:

- A reaction or response to a particular process or activity.
- Evaluative information derived from such a reaction or response.
- Knowledge of the results of any behaviour, considered as influencing or modifying further performance.

These definitions clearly express the core meaning of feedback: to give back to an individual information about something they did. Research has attempted to identify the concept of feedback within specific disciplines. Herold and Greller (1977: 142) identified that management literature “deemed [feedback] central to issues of training, performance, motivation and satisfaction” but that “little empirical effort has been expended in an attempt to understand it”. They continued by summarising that “the exact meaning of the [feedback] dimension remains highly uncertain” (ibid: 142), however their findings did not lead to a definition of feedback, but offered foundations for further research. Later, again in the management discipline, Ramaprasad (1983: 4) noted that “there is little consensus among management theorists on the definition of the concept”, and then defined feedback as the “information about the gap between the actual level and the reference level of a system parameter which is used to alter the gap in some way” (ibid: 4). In dynamic psychotherapy, Berger (1994) reported that “the literature of psychoanalysis and dynamic psychotherapy shows little understanding or use of the concept of feedback", and then disassembled feedback in psychoanalysis and identified a "systemic process" of feedback in psychotherapy (ibid: 235).

Recently, van de Ridder *et al.* (2008) investigated feedback in clinical education. They noted that "The term feedback is now used and interpreted in many different ways" (ibid: 190) and that within clinical education "there seems to be little consensus about its definition" (ibid: 190). Van de Ridder *et al.* (2008) identified that feedback was not a singular construct, but rather was the combination of several characteristics. After reviewing previous research and definitions of feedback, they identified nine relevant characteristics:

- Content – the information to be conveyed.
- Aim – the intention of the feedback.
- Feedback recipient – the person receiving the feedback.
- Form – how the information will be sent.
- Preparation – prior to sending the feedback, what must be done.
- Source – internal or external information.
- Feedback provider – the person giving the feedback.
- Communication conditions – timeliness and directness.
- Contextual factors – the place where the feedback is received.

Although feedback is a factor present in several theories of motivation (see section 2.2), the definition and use of feedback varies across the different theories. All of the theories of motivation use feedback as a tool for providing individuals with the knowledge of their performance (Job Characteristics Theory, Goal Setting Theory), with knowledge of the results of their actions (Achievement Theory), or with recognition (Hygiene Theory). While feedback is identified by all four theories of motivation, there is minimal discussion on the importance of feedback other than to provide information, and very limited identification of feedback characteristics and the impact that they may have. This suggests that the literature has not adequately considered the possible impact that the characteristics of feedback may have.

2.4 FEEDBACK IN SOFTWARE ENGINEERING

The literature investigating motivation in software engineering does not provide a clear understanding of feedback in software engineering. While some research has investigated feedback in software engineering, the majority of what we know is from the results of studies focusing on something other than feedback. The literature is often investigating a broader subject like "what motivates software engineers" or is trying to understand the relevance of a motivation theory in software engineering, and feedback is included as part of that theory. Previous studies have typically identified feedback as a factor affecting motivation in software engineering while investigating the suitability of a theory of motivation that includes feedback (examples: LeDuc, 1980; Couger and Zawacki, 1980; Gambill *et al.*, 2000). No previous study was found that focused solely on investigating the definition and impact of feedback in software engineering.

Due to the lack of research focusing on feedback, it is unsurprising that there is no literature that attempts to identify, distinguish, and investigate the impact of the different characteristics of feedback. Kim and Wright (2008) investigated the impact of performance feedback on employees' work exhaustion, and they summarised that "providing employees with more feedback about their work" was an "important way" supervisors could lower their employees' work exhaustion, and in turn lower employee turnover intentions. Zawacki (1992) identified that "feedback from managers" was "the most important need of IS professionals" (ibid: 74) when summarising findings from several previous studies spanning over 14 years (examples: Couger and Zawacki (1980), Dittrich *et al.* (1985), Couger and Adeslberger (1988)). Zawacki (1992) argued that the "most critical personnel issue" of that decade was to keep programmers motivated.

Although there has been limited research specifically investigating feedback, the results of investigations that include feedback suggest that feedback is a factor affecting the motivation of software engineers. In 1984 Cheney (1984), a management researcher in Georgia, USA, was investigating the effects of individual characteristics, organisational factors and task characteristics on programmer productivity and satisfaction. Cheney identified a strong and important need to provide programmers with adequate feedback:

> *"[Programmers] need feedback both for guidance and to satisfy their psychological needs with regard to performance. If they do not obtain this from their direct supervisors their productivity and satisfaction will suffer." (ibid: 213)*

More recently, Carayon *et al.* (2003), while investigating the role of gender in the IT workforce, analysed questionnaire data collected during 1999 to 2001, and concluded that:

> *"The most consistent predictor of job satisfaction is feedback" (ibid: 61).*

While investigating collaboration practices in global inter-organisational software development projects, Paasivaara and Lassenius from Helsinki University characterised the use and importance of feedback:

> *"Feedback received from the customer also gave the subcontractor confirmation that the correct tasks were being performed. Getting feedback also motivated team members." (Paasivaara and Lassenius, 2003: 193)*

Kim and Wright (2007) investigated IT employee work exhaustion. They highlighted "performance feedback" specifically, and how it had an indirect effect on work exhaustion:

> *"Performance feedback had important indirect effects on work exhaustion by increasing role clarity and perceived advancement opportunities" (ibid: 161).*

Chen (2008), an information management researcher in Taiwan investigated job satisfaction among IS personnel, and identified three key factors to increasing the job satisfaction of IS personnel:

> *"Jobs with the features of feedback, professionalism and autonomy can most easily increase the job satisfaction of IS personnel" (ibid: 105).*

These studies all identify the potential impact and importance of feedback while investigating motivation in software engineering, and they highlight particular aspects of factors relevant to feedback, motivation and job satisfaction – yet without providing a clean and complete account of feedback.

2.5 LITERATURE SUMMARY

The definition of feedback in theories of motivation is incomplete, and feedback remains an underexplored factor that can affect software engineers. The theories of motivation that identify feedback as a factor do not consider the possibility and impact of different forms of feedback, and instead identify feedback just as a method to provide the recipient with the knowledge of the results of their actions.

While feedback has been identified as a factor that can affect the motivation of software engineers, it remains unclear how feedback affects software engineers, and if different forms of feedback can alter the effect of received feedback.

Neither motivation theories nor research in software engineering define any characteristics of feedback, which have been identified in clinical education (van de Ridder *et al.* (2008)). No studies were found that focused exclusively on feedback in software engineering. Research that identified feedback as a factor impacting the motivation of software engineers were typically investigating the applicability of a motivation theory that included feedback (examples: LeDuc (1980); Couger and Zawacki (1978); Gambill *et al.* (2000)).

The literature review illuminates a gap in knowledge about feedback, its characteristics, and its impact in software engineering. Theories of motivation do not properly consider the impact of the different forms of feedback, and previous research investigating motivation in software engineering has not focused on feedback or any feedback characteristics. As part of the larger study of motivation in software engineering, we need to better understand feedback, the characteristics of feedback, and the impact that feedback and the different values of feedback characteristics can have on the motivation of software engineers.

CHAPTER 3

FEEDBACK SCOPING STUDY

Before any attempt is made to investigate the impact of feedback on the motivation of software engineers, feedback must be identified and defined in software engineering environments. This chapter presents a preliminary study to investigate feedback and how feedback is described by experienced software engineers. The feedback characteristics identified in clinical education provide a suitable comparative foundation for characteristics identified during this study. This study aims to identify the relevant characteristics of feedback that occur in software engineering environments, and compare any identified characteristics to the characteristics of feedback identified in clinical education.

The following sections discuss the research design (3.1), present the participants (3.2), describe the analysis approach (3.3), present the findings (3.4) discuss the threats to validity (3.5), and discuss the findings (3.6). Finally, the state-of-play in the terms of the progress of this research is presented including the knowledge from the findings of this study (3.7).

3.1 RESEARCH DESIGN

Before any major study could be designed, feedback in software engineering and the characteristics that define feedback in software engineering had to be investigated. The most concrete characteristics came from clinical education; it was unclear whether it would be valid to assume that software engineering had the same characteristics as those identified in clinical education. The potential differences between feedback characteristics found in clinical education and those found in software engineering environments had to be evaluated before feedback could be investigated as an individual factor affecting the motivation of software engineers.

Our understanding of feedback in software engineering environments is limited, and requires the collection of rich data using a flexible, adaptable research method. The research is exploratory and is investigating a topic that is not clearly identified nor defined. Explorative research benefits from a research method that allows flexibility at the point of data collection for both the participant and the researcher, and enables emergent themes to be identified and explored in-person.

Therefore, a semi-structured interview was designed with the intention of eliciting from software engineers how they defined feedback in software engineering. A well-designed interview can prompt the participant to reflect and to respond to questions with rich and comprehensive answers, as well as providing the researcher with the flexibility to ask questions opportunistically in real-time as a response to something the participant said. The interview instrument used during this study is in appendix A section 1.

The participants were given an oral and written introduction to the research and to the aims of this study (appendix A section 2). After the introduction the participants completed a demographic questionnaire (appendix A section 3). The participants received a consent form and were asked to date and sign it if they consented to the data they provided during this study being used for academic purposes (appendix A section 4). Ethical consent was obtained from the Open University's Human Research Ethics Committee (appendix B section 3). Participants were then asked if they were willing for the interview to be audio recorded to aid data analysis, to which all participants agreed. After the introduction and the demographic information form, the interview was divided into four themed groups of questions, discussed in the following paragraphs:

Theme One – Work History. Participants were asked additional questions to the demographic information form about their work history. This was to ensure that they had not recently changed their role or job, which could have led to the demographic questionnaire providing an unrealistic representation of their work history as the demographic questionnaire collected data about recent projects only.

Theme Two – Interpretation and Understanding of Feedback. Participants were asked about their interpretation and understanding of feedback, what the term 'feedback' meant to them. The participants were then asked to provide examples of feedback they had received, and to identify what they thought the characteristics of the feedback were.

Theme Three – The Impact of Received Feedback. Participants were asked to discuss the impact of feedback they had received, and examples collected from the previous theme's responses were used to prompt the participants to recall the impact of feedback they had already discussed.

Theme Four – Reason for Employment Change. Participants were asked to discuss why they changed from their previous job to their current one, and if the change was in any way related to the feedback they had received.

Each interview took 20-30 minutes to complete.

3.2 PARTICIPANTS

The participants (Table 3.1) all had previous experience as software engineers. The participants were aged between 30 and 50, and had experience in different software development environments, working in teams of between 5 and 30 people.

TABLE 3.1 – PARTICIPANT DEMOGRAPHICS

Name:	Gender:	Level of Education:	Development Experience	Years' Experience:	Age:
P. 1	Male	Doctorate	Waterfall	2 years	31-40
P. 2	Male	Doctorate	Agile	5 years	41-50
P. 3	Male	Doctorate	Spiral	10+ years	31-40

3.3 DATA ANALYSIS

The data analysis was inductive, identifying emergent themes from the participants responses to the non-demographic information questions.

Two types of data were collected during the interview. The first was demographic information collected through demographic questionnaires given to the participants at the start of the interview. This data was extracted from the completed forms and tabulated.

The second type of data was the audio recordings of the participants' responses to questions during the interview. The audio recordings for each of the interviews were transcribed verbatim.

The transcripts were divided into segments, and each response was coded in relation to the question, any characteristics of feedback identified by the participants, and any themes emerging directly from the participants' responses. This coding enabled the analysis to identify the characteristics of feedback in software engineering as reported by the participants. Feedback characteristics were identified from the participants' responses.

3.4 FINDINGS

During the interviews the participants defined and discussed what they considered feedback to be, how feedback happens in software engineering environments, and the possible impact feedback can have on software engineers. The participants identified and discussed ten different characteristics of feedback, shown below:

- Source
- Goal
- Medium
- Direction
- Instigation
- Setting
- Timeliness
- Content
- Preparation
- Feedback recipient

During the interviews the participants provided examples of feedback that could be characterised by eight of the ten feedback characteristics. These examples (Table 3.2) were collected to ensure that the interpretation of the named characteristic was correct, and to explore the possible values of each feedback characteristic. The participants identified examples that were typical of feedback they had received in software engineering environments, using previously-discussed received feedback as the initial examples. Each identified example emerged from the responses of the participants, with each example being reported by at least one participant.

TABLE 3.2 – EXAMPLE VALUES OF IDENTIFIED FEEDBACK CHARACTERISTICS

Source	Goal	Medium	Direction	Instigation	Setting	Time	Content
Subordinates	Get someone promoted	Verbal	From others to me	End of project	Meeting	ASAP	Performance on task
Manager	Fulfil promotion criteria	Written report	From other to me	Issues arising	Casual	Later	Performance on Project
Customer	Reduce stress	Body language	From others to my team	Prior decisions	Written report		Attitude
Development Team	Reassurance	Email	From other to my team				Quality of work
Developers	Remove tension						

Motivation, behaviour, job satisfaction, and feelings were all identified by the participants as aspects that could be affected by received feedback. One participant also discussed the effect of the source of the feedback and how the feedback message is conveyed to them, reporting that “some types of feedback depend on who is delivering it and how it’s delivered”. The identified aspects were discussed by the participants, including examples of how feedback could affect each aspect, as discussed below:

Motivation. The participants provided examples of how their motivation could be affected. One participant discussed how receiving positive feedback from a colleague “would motivate me” and another participant reported that negative client feedback could “sap one’s motivation to carry on”. Receiving negative client feedback might have a long-term effect on the participant’s motivation “you might then potentially have a week where you do the bare minimum to get by on a project because you just don’t feel motivated enough to deliver above and beyond”.

Behaviour. The participants discussed the impact that feedback could have on their behaviour. One participant reported that he needed feedback to reinforce and confirm his behaviour, claiming that “you may need someone to tell you you’re doing fine so you know actually it’s worth carrying on with” and that “if you know that’s fine you’ll continue doing it”, but without this feedback, the required behaviour may be discontinued: “you might be doing something and you might think well no one’s said anything about this, maybe they don’t want me to do this, maybe I should stop doing this.”

Job Satisfaction. The participants discussed how receiving feedback could affect their job satisfaction. Feedback received from a code compiler could lead to the participant feeling a “sense of satisfaction” if his code compiled and did everything he wanted it to do, but also could lead to the participant feeling a “sense of dissatisfaction” if his code failed to compile or didn’t do what he wanted it to do.

Feelings. The participants reported that feedback could affect their feelings. Positive feedback from their colleagues could be “very reassuring” and could help in “reducing stress”, and also reduce any “tension” in the team. Positive feedback may also lead to a participant feeling “quite good” about themselves. One participant reported that the lack of feedback could also have an impact on his feelings, and reported feeling “quite stressful” when not receiving feedback as he did not know if he was “doing the right things”.

3.5 THREATS TO VALIDITY

The findings of this study are subject to four main threats to validity. First, the participants may not have been representative of software engineers. This threat to validity is limited both by the design and the focus of the study. This study sought to investigate an under-explored area of research, so an initial investigation with experience software engineers was an appropriate starting point.

Second, the participants’ memory of their past experiences of feedback may have been distorted. This threat to validity is limited by the design of the study, focusing on recent experiences, and asking the participants to speak in general and non-specific terms of their experiences.

Third, the sampling of these participants was opportunistic, and not random, which may bias the findings as the participants may not be a representative sample of software engineers. The number of participants limits the possible representativeness of the findings, meaning that these findings cannot be viewed as representative, but as indicative.

All three of these are threats to the external validity of the findings.

Fourth, there is the potential for bias in the design of the study. The study was designed to explore how software engineers discuss feedback, and was designed by a software engineer. This potential threat to validity is limited as the designed study was piloted and reviewed by two experienced researchers and three post-graduate research students.

3.6 SUMMARY

The feedback characteristics identified by the participants used different names compared to the characteristics identified in clinical education. The participant's used terms to identify feedback characteristics that sometimes conflicted with the definition of the same term in clinical education. For example, the participants defined the 'source' as the person providing the feedback, whereas the clinical education feedback characteristics define the 'source' as the cause of the feedback, such as the results of the actions of the person receiving the feedback. Table 3.3 shows the relationship between the feedback characteristics identified by the participants of this study and the feedback characteristics identified in clinical education.

TABLE 3.3 – IDENTIFIED FEEDBACK CHARACTERISTICS

Participant-Identified Characteristics			Literature-Identified Characteristics
Source	←	→	Feedback provider
Goal	←	→	Aim
Medium	←	→	Form
Direction	←	→	Communication conditions
Instigation	←	→	Source
Setting	←	→	Contextual factors
Timeliness	←	→	Communication conditions
Content	←	→	Content
Preparation	←	→	Preparation
Feedback recipient	←	→	Feedback recipient

Eight of the ten identified characteristics shown in Table 3.3 were explicitly named by the participants. The other two feedback characteristics shaded in blue, preparation and feedback recipient, were discussed but not named explicitly. The participants were asked to focus on their experiences of receiving feedback, not providing it, during the interview. This focus caused the preparation and feedback recipient characteristics to be discussed, but not named directly by the participants while they focused on their experiences of receiving feedback.

The example values of the feedback characteristics discussed by the participants provide a foundation for identifying different forms of feedback in software engineering. The different forms of feedback can be identified by combining the specific values for each feedback characteristic. Subsequent empirical studies in this research (chapters 5, 6 and 7) investigate and identify common forms of feedback occurring in software engineering environments, and evaluate if there is a variation in the impact of different forms of feedback.

The participants reported that feedback could affect their behaviour, motivation, job satisfaction and feelings. Job Characteristics Theory (Hackman and Oldham, 1976) identified low absenteeism and turnover (change in behaviour), high internal work motivation (motivation), and high satisfaction with work (job satisfaction) as possible outcomes from any job that scored highly in five different job characteristics that included feedback. The direct relationship between feedback and its impact on the aspects identified in Job Characteristics Theory (Hackman and Oldham, 1976) is investigated further in the other empirical studies of this research (chapters 5, 6 and 7).

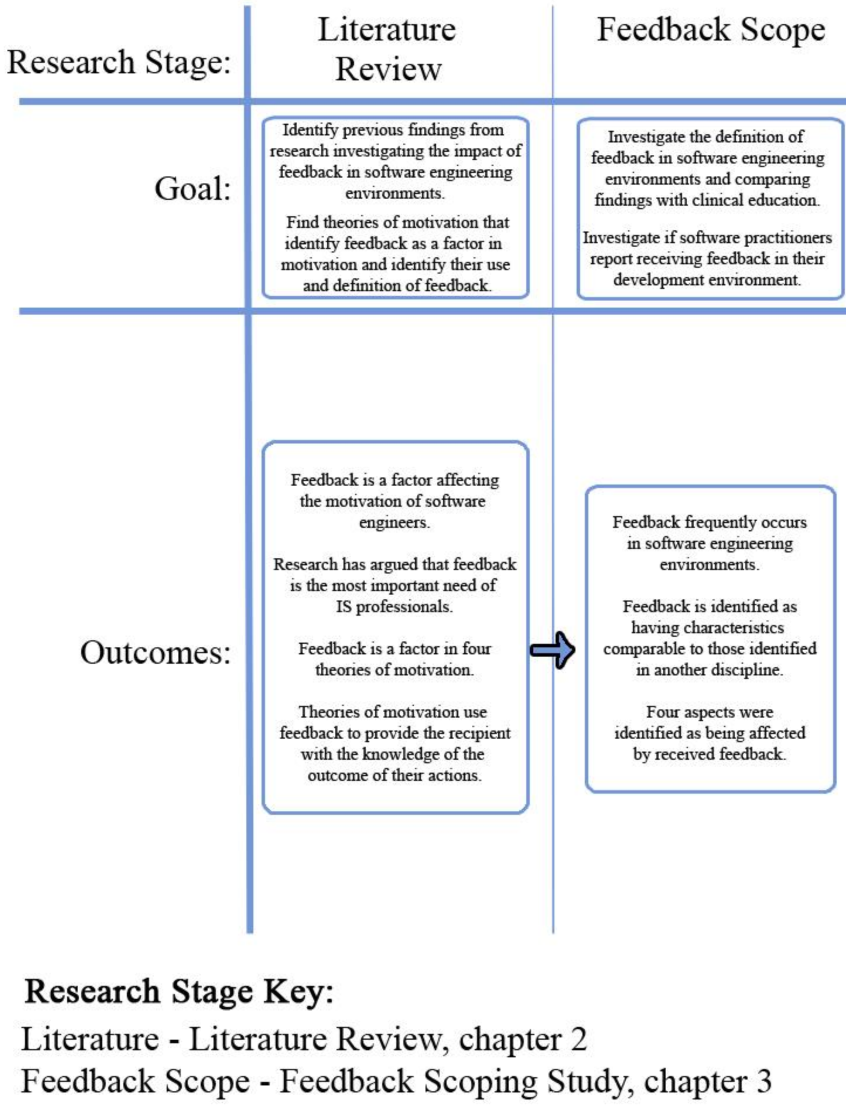
3.7 SO WHAT DO WE KNOW NOW?

The findings of this study indicate that software engineers reported feedback as present in software engineering environments and that their definition and characterisation are comparable to those from clinical education. The participants provided responses to questions in a semi-structured interview that identified ten characteristics of feedback, which related to those identified in clinical education.

The example values (Table 3.2) for each feedback characteristic provide the foundation for investigating different forms of feedback, and for evaluating the importance of different forms of feedback and different feedback characteristics. The aspects listed as being affected by feedback require further investigation to address the relationship between an identified instance of feedback, and its effect on behaviour, motivation, job satisfaction and feelings.

These findings provide a useful and appropriate starting place to use these feedback characteristics and identified aspects as the foundation for a larger study investigating the impact of feedback, the importance of the different feedback characteristics, and the affect that the different values of a feedback characteristic may have in software engineer environments.

A map of the research progress at this point is shown in Figure 3.1.



Research Stage:	Literature Review	Feedback Scope
Goal:	Identify previous findings from research investigating the impact of feedback in software engineering environments. Find theories of motivation that identify feedback as a factor in motivation and identify their use and definition of feedback.	Investigate the definition of feedback in software engineering environments and comparing findings with clinical education. Investigate if software practitioners report receiving feedback in their development environment.
Outcomes:	Feedback is a factor affecting the motivation of software engineers. Research has argued that feedback is the most important need of IS professionals. Feedback is a factor in four theories of motivation. Theories of motivation use feedback to provide the recipient with the knowledge of the outcome of their actions.	Feedback frequently occurs in software engineering environments. Feedback is identified as having characteristics comparable to those identified in another discipline. Four aspects were identified as being affected by received feedback.

Research Stage Key:

Literature - Literature Review, chapter 2

Feedback Scope - Feedback Scoping Study, chapter 3

FIGURE 3.1 – RESEARCH PROGRESS MAP

CHAPTER 4

RESEARCH DESIGN FOR THE FEEDBACK IN ONE SOFTWARE ENGINEERING ENVIRONMENT STUDY

Theories of motivation identify feedback as a factor used to provide the recipient with knowledge of the outcome of their actions or recognition. Theories of motivation do not identify differences between instances of feedback, nor a method to identify different forms of feedback. Research in clinical education has investigated feedback and identified and defined different characteristics of feedback. Findings from interviews with experienced software engineers (chapter 3) identified that comparable feedback characteristics existed in software engineering environments, and that this definition provides a useful and appropriate starting point to investigate feedback in software engineering.

Building on the findings of the Feedback Scoping Study (chapter 3), this study aims to investigate feedback in software engineering, identifying different forms of feedback found in software engineering and investigating the impact the identified feedback can have on software engineers. Feedback Scoping Study (chapter 3) identified that feedback is discussed by software engineers as having characteristics comparable to those identified in clinical education, and that feedback can affect four aspects of software engineers. This study builds on these findings by investigating if the identified feedback characteristics can be used to identify different forms of feedback found in software engineering, and investigating how any identified different forms can affect software engineers.

Feedback, a factor affecting motivation, is internally experienced. Feedback is not something that is typically discussed by software engineers, and its impact is not always externally visible. Investigating feedback, how it is perceived by software engineers, and the impact it can have requires direct elicitation from software engineers about their day-to-day experiences of feedback.

The following sections describe the foundation of the research design (4.1), discuss the research methods used during this study, including a semi-structured interview (4.1.1), a diary study (4.1.2), and direct observation (4.1.3), discuss the use of a personality inventory during the research (4.2), present the design of the data analysis (4.3), provide a summary of the collected data (4.4), identify the relevant ethical considerations (4.5), and discuss how the research methods combined to complement each other and this research (4.6).

4.1 RESEARCH DESIGN

The research design uses an empirical approach, defined by Black (1999: 3) as “information, knowledge and understanding gathered through experience and direct data collection”. The decision to use empirical research methods was taken with the goal of gaining a rich understanding of feedback in software engineering environments. Lehman and Belady (1976) and Harrison *et al.* (1999) argue that the nature of empirical research causes it to more fully reflect the environment being studied than other research approaches, supporting the suitability of an empirical approach for this research as environmental factors need to be considered.

To ensure that all potential aspects of feedback are considered, a wide scope of investigation is required. The research approach must view feedback from different perspectives and at different points in time. The research methods should be flexible and adaptable so that emergent data can be engaged with and investigated at the point of data collection.

The requirements of this study support the use of a research design that uses multiple research methods in the form of method triangulation to investigate a single phenomenon. Cohen and Manion (2000: 141), while discussing research methods in education, define triangulation as "attempt to map out, or explain more fully, the richness and complexity of human behaviour by studying it from more than one standpoint." Altrichter *et al.* (2008: 147) argue that the use of triangulation "gives a more detailed and balanced picture of the situation." Method triangulation is used in this research design to ensure the data collected from the participants is representative of what actually happens, and to provide a rich encapsulation of feedback in software engineering.

The suitability of the selected research methods is found both in the type of data likely to be collected, and in each method's ability to be flexible when required by the researcher or the participants. The collected data combined the perceptions and reflections of software engineers, and the research methods were open and flexible to exploring emergent themes and topics of discussion. This research studied software engineers in the field, capturing data from their reflections and as it happened.

Feedback, the definition of feedback, and the effect of received feedback is not something believed to be discussed during a normal working day in software engineering environments, requiring this research to use direct question-driven elicitation from software engineers. The research methods were used to collect data investigating feedback in software engineering, but due to the nature of each individual research method, they collect different types of data investigating feedback at different granularities.

SEMI-STRUCTURED INTERVIEW

Semi-structured interviews were used to collect software engineers' reflective opinions on factors which motivate and de-motivate them. Feedback is investigated indirectly by questions asking about how they communicated in their environment and how they would know when they had done a good or bad job.

Semi-structured interviews were chosen as they allow a planned, structured approach while also providing the researcher with the flexibility to gain clarification on something the participant may have said. Semi-structured interviews also allow the interviewer to ask further exploratory questions in reaction to the participant's responses to any of the questions.

Interviews can be used to collect rich data in the language of the participants and to provide real-time responses to questions. The interview method allows the researcher the flexibility and adaptability that is required when collecting data about a phenomenon which lacks substantial previous research.

DIARY STUDY

The diary study collected data from participants at two points in time: once whenever they received something they perceived as feedback, and once at the end of each working day. This data was used to collect more detailed, specific instances of feedback, and to investigate how received feedback affected them at the time of receiving the feedback, and at the end of each working day.

The diary method was selected as it enables the participants to record events as they happen, and provide realistic accounts of events either directly after they occur or within minutes.

Alaszewski (2006) argues that when investigating multiple events that occur within the day, diaries are typically more accurate than other methods. Findings from the Feedback Scoping Study (chapter 3) found that feedback happens often in software engineering, and recording instances of feedback as they occur in software engineering requires a method that is flexible and allows self-reporting at different points during each day.

Lazar *et al.* (2009) identified three different types of diaries:

- Elicitation Diary
- Feedback Diary
- Hybrid Feedback and Elicitation Diary

An elicitation diary is used to record data which will mainly be used to prompt a later method, such as an interview. The data collected is basic information about the event of interest, and the researcher uses the data either to prompt further research or during a future research method. A feedback diary is used as the main form of data collection during the study, where the participants are asked to record more detailed information that is the main collection of data during the research, and no other research methods are used.

A hybrid feedback and elicitation diary joins the characteristics of both the feedback and elicitation diaries. A hybrid diary collects data that is useful by itself, and that is also useful in informing later research methods. This research will be using a hybrid diary, collecting data from participants on feedback instances as they occur, but also using the data to prompt further questioning.

PARTICIPANT OBSERVATION

Participant observation allowed for a better, deeper understanding of the environment in which the software engineers worked. Cross-comparison between observed instances of feedback and recorded instances of feedback collected during the diary study was undertaken.

PERSONALITY INVENTORY

The participants were asked to complete an online presentation of the five-factor personality inventory. The personality inventory was included in the research design to offer a point of comparison between the participants, and to ensure that the participants were representative of the wider community of software engineers. A comparison of the collected personality results with the personality inventory results from other software engineer studies established the representativeness of the participants of this study, as discussed further in section 5.5.6.

TABLE 4.1 – RESEARCH DESIGN SUMMARY

Research Method	Implementation	Used to
Semi-structured Interviews:	Direct 1-to-1 interview with each participant	Identify what (de)motivates the participants, how they communicate, and how feedback happens in their environment
Diary Study:	1 week-long diary study	Identify what feedback the participants receive, how it affects them at the time of receiving it, and how it affects them at the end of each working day
Participant Observation:	1 week-long observation of participants	See feedback occurring, and gain a rich understanding of the participants' working environment
Personality Inventory:	100-question personality inventory	Possible future analysis avenue, and to ensure that the participants were representative of software engineers

PARTICIPANTS

This study required participants who would engage with the research and willingly provide their time and commitment. The choice of participants was limited by the design and selection of the research methods, requiring participants who were willing and able to engage with the study in different research methods during multiple stages of research. The design of the research required a team-based environment, with access to multiple teams of willing software engineers.

Red Gate, a software development company based in Cambridge, UK (see section 5.2 for an extensive description of Red Gate and the participants working environment) was identified as a suitable company with participants suitable for this research. The software engineers at Red Gate worked in an agile environment, which encourages face-to-face communication[2]. Red Gate had teams of developers working together on team-specific-projects every day, supporting the probability of being able to investigate, record, and witness feedback as it occurs in a software development environment. A full description of the participants, including their development experience and background information can be found in section 5.3.

RESEARCH DESIGN SUMMARY

A summary of the research design is shown below in Table 4.1. To summarise, the research design for this study is focused on eliciting the opinions of active software engineers using a combination of research methods that allow data to be collected on both motivation and feedback over time.

The research methods combined to produce a picture of feedback as reported by software engineers, containing reflective accounts of feedback previously received and its impact, time-of-receipt reports of individual instances of feedback, and end-of-day reflections on the impact of feedback received and reported during that working day. These reports of feedback are complemented by observations of the environment which aids contextual understanding, and participant demographics and personality inventory results which aids data analysis.

TABLE 4.2 – RESEARCH METHODS TIMELINE

Week 1 & 2	Week 3
Semi-structured interviews - The participants were interviewed individually over a two week period Diary study introduction - The participants were invited to attend a brief introduction to the diary study	Diary study - The participants completed a one-week diary study Observation - The participants were observed during the one-week diary study Personality inventory - The participants received directions during this week instructing them how and where to complete the personality inventory

[2] http://agilemanifesto.org/.

This study was completed as described in the timeline shown in Table 4.2. The separate research methods were ordered so that the participants were introduced to the research and the research focus during the interview, were given a brief introduction to the diary study, and were more prepared and able to complete the diary study due to having a better understanding of the aims and focus of this research. During the diary study the participants were observed, allowing the possibility of exploring feedback witnessed during observation but not reported in the diary study. Each of the research methods used in this study is discussed in detail in the following sections.

4.1.1 SEMI-STRUCTURED INTERVIEW

INTERVIEW INSTRUMENT DESIGN

The interview was designed with the core focus of encouraging software engineers to discuss motivation and feedback in their current working environment. The Feedback Scoping Study (chapter 3) and pilot studies helped shape the composition of the interview. The questions were open-ended and allowed the participants to reflect and respond to each question with as short or as long an answer as they wanted.

The decision to use a semi-structured exploratory survey approach was both to allow the creation of a set of questions that emerged as appropriate from previous studies, and to provide the researcher with the flexibility to be able to ask other questions during the interview if the situation warranted them, while investigating a phenomenon that is not fully understood in this domain. Oppenheim (2000) describes an exploratory interview as one where the research is “concerned with trying to understand how ordinary people think and feel about the topics of concern to the research” (Oppenheim, 2000: 67).

The design of the interview was intended to investigate feedback and motivation without directly introducing the terms during the interviews, limiting the potential bias from participants reacting to words that have a meaning attached to them, which may not be the same as the intended investigation.

The interview protocol (appendix B section 1) was divided into three sections: demographic information, motivation in software engineering, and feedback and feedback impact in software engineering. The first section (demographic information) is of multiple-purpose, as it collects the demographic details of the participants, and initiates conversation and builds rapport with the participants and the researcher. The demographic questions provide a relaxed opening to the interview, and help the participants to engage with the researcher early on with basic questions requiring little thought or reflection.

After the demographic questions, the participants were asked questions investigating motivation and de-motivation, including “What encourages you to go that extra mile at work?” and “Is there something that really saps your energy at work?” This phrasing was used to avoid preconceptions that can come with terms such as motivation and de-motivation.

The final section of the interview focused on feedback and feedback impact in software engineering. This section took a similar approach to the previous section by asking open-ended questions without using the term feedback. The participants were first asked what the communication was like in their team, and how often team members spoke to each other. Further questions investigated how they knew when they had done a good or bad job, and how this made them feel. The full interview instrument used is in appendix B section 1.

INTERVIEW CONDUCT

The interviews were held at the participants’ current working location. The interviews were conducted one-to-one between participant and researcher. The participants were originally contacted by their manager, who asked them to participate. Those that were willing to participate agreed a day, time and location with their manager and the researcher.

Prior to the start of the interview, the participants were each sent a copy of the consent form via email, and all participants agreed, signed and returned the consent form at the start of the interview. The consent form included an introduction to the research (appendix B section 2). The interviews were recorded, after each participant was asked for permission. Notes were taken during each interview for recording of information for follow-up questions, and to indicate important responses to be noted for analysis.

Each interview lasted between 18 and 80 minutes. At the end of the interview the participants were given a brief overview of the diary study and asked if they would be willing to participate in the diary study. Those who did not wish to participate in the diary study were informed that they would be getting an email with directions of where to complete the personality inventory, and those who were willing to participate in the diary study were informed that they would receive an email with further instructions including directions about where to complete the personality inventory,.

In total, 24 people who identified themselves as either full or part-time software engineers were interviewed. Participants who considered themselves part-time software engineers were both software developers and project managers, spending between 20-50% of their time doing software engineering tasks. The recordings of the interviews compose over 15 hours of audio.

INTERVIEW LIMITATIONS

With any research method there are limitations. With the interview, the data collected relies heavily on the engagement of the participants, and the skill of the researcher asking the questions. The interview protocol was piloted, and all the participants were willing to participate, and while participation was encouraged, the participants were never mandated to participate; piloting the interview provided the researcher with invaluable experience of collecting data using the interview instrument, reducing the risk of the interviewer's skill affecting the data collected in the interviews.

One limitation of the interview research method is the quality and accuracy of the questions being asked. To ensure the questions were appropriate and elicited the desired type of responses from the participants, the research questions were piloted and modifications were made to the interview instrument and protocol as required. During the piloting of the research, valuable experience of conducting interviews was gained, improving the skill of the researcher.

4.1.2 DIARY STUDY

DIARY INSTRUMENT DESIGN

The diary instrument was produced with three design goals:

1. Capturing instances of feedback.
2. Capturing the daily impact of received feedback.

3. Enabling participants and researchers to view submitted data in real time. This required careful planning and consideration in order to ensure that the correct data was collected from the participants.

To meet the first design goal, a form was designed to allow the participants to record instances of feedback they received during the study. The form allowed the participants to unpack the feedback they received by providing values of several feedback characteristics as identified in the Feedback Scoping Study (chapter 3) and from the reviewed literature (chapter 2). The feedback characteristics included pre-generated value options, as well as additional free-text room to ensure the participants had enough space to report different values for the feedback characteristics.

To meet the second design goal a form in the diary was created that asked the participants to report a summary of how they felt at the end of a working day (day summary). The form enabled participants to record how they felt at the end of each day with free-text space. They were asked whether and how the feedback they had received had affected them at the end of each working day.

To meet the third design goal, the online diary was designed to output data to the participants. Once an instance of feedback was submitted, the participants were able to view and edit it. When the participants completed a day summary, all of the instances of feedback they had submitted during that day were shown for their reference. The researchers were given access to an overview of all of the data submitted by the participants, enabling investigation during the diary study.

ELECTRONIC PRESENTATION

A literature review by Hufford and Shields (2002: 52) claimed that “The empirical literature reflects a clear subject preference for electronic diaries relative to paper diaries”. Stone *et al.* (2002) compared the number of participants completing an electronic or paper based diary in their study investigating compliance rates in diaries of adults suffering from chronic pain. They found that while 90% of participants claimed they had completed the paper diary, only 11% actually had, compared to 94% of participants who completed the electronic diary in full. In another publication (Hufford and Shields, 2002: 52) the same researchers reported that “subjects using electronic diaries rated them just as easy to use, read, and carry with them as paper diaries”.

All software engineers are familiar with computers and the internet, and will have access to a computer during their working day, suggesting that the participants for this research will be accepting and willing to complete an electronic diary. And electronic presentation of the diary was chosen as it is the most suitable for the participants, and the reviewed literature suggests that a higher completion rate will be achieved using an electronic diary.

WEB-BASED VS. SOFTWARE-BASED

Upon designing the diary it was possible to produce a web-based or computer-based diary. Due to access limitations and ease of installation, a web-based electronic diary was chosen as the most suitable and practical choice for this study. This allowed easy access to the participants' responses, enabled maintenance of the system remotely should any problems arise, and allowed the participants to access the diary on any web-enabled electronic device.

THE DIARY

The diary was programmed in PHP and MySQL, using HTML and JavaScript. The diary website was hosted on a secure server, and the participants were each given a unique username and password with which to access the diary, as well as the ability to change their password. This screen provided the users with information about what data they had submitted, and what data they still had to submit, as well as providing them with an FAQ section with useful information and clarification. Screenshots of the diary system can be seen in appendix B section 7.

By navigating around the web-based system, the participants were able to submit feedback instances, update feedback instances (the same day as submitted), submit day summaries, update day summaries, and view submitted feedback instances and submitted day summaries. The participants were able to select responses to questions from a pre-generated list, as well as selecting "Other" and indicating an alternative response to any question in a free-text box. This facilitated both participants who wished to provide short concise responses that fitted within the pre-generated list of options, and participants who wished to expand on their responses or indicate an option not in the pre-generated list of responses.

DIARY ADMINISTRATION

Each diary study participant was emailed the URL address of the diary site, a unique username and a unique password to access the diary, and instructions on how to submit a personality inventory. The diary study participants were all invited to attend an introduction to the diary in person at their offices, and were allowed to ‘play’ with the diary for three days prior to the start of the diary study so that they could familiarise themselves with the diary and ask any questions prior to beginning the diary study.

The diary study lasted between 4-5 days for each participant, during one working week, depending on their availability. During the first day, participants who had not input any feedback instances by 4:30pm were automatically emailed to prompt their input. During each subsequent day, the participants received two email prompts: (1) one email at 9:30am if they had submitted no instances of feedback on the prior day or if they had not submitted the day summary from the previous day, and (2) one email at 4:30pm if they had not submitted any feedback instances in the diary during that day.

Participants were also prompted to complete the personality inventory once during the diary study, as they had already received instructions. The gentle prompting was used to remind the participants to use the diary, as many of them were often focused on their work and forgot, and used their memory to report on feedback they had received previously during the day.

Behind the user-completed forms the diary collected other important information: the date and time of each submitted feedback instance or day summary. The diary also recorded when a change was made to a previously submitted feedback instance or day summary, and the details of the change that was made. This was important as it ensured that we could monitor if the participants changed their original submissions, and if so, what they changed. The diary also asked the participants to record the time when each feedback instance was received, and this allowed us to know how soon after receiving an instance of feedback the participants recorded it in the diary.

In total, 15 of the 24 interview participants agreed to take part in the diary study. During the 5-day diary study, 45 completed day summaries and 76 instances of feedback were collected. 6 participants completed a full week, submitting 5 day summaries, or 4 if they were away for one day during the study (2 of the 6). 6 participants submitted 5 or more instances of feedback each during the diary study.

DIARY ANALYSIS DESIGN

The type of data likely to have been collected in the diary study influenced the relevant and appropriate analysis approach for this research method. Due to the design of the diary, the analysis was split into two sections: (1) tabulation of pre-listed responses, and (2) content analysis of user submitted data, specifically instances of feedback and day summaries.

DIARY LIMITATIONS

One limitation of using the web-based diary was the reliance on user-driven data input. While prompting emails could be sent, if the participants did not take the time to complete the diary, then enough data would not have been collected. Similarly, the design of a diary study relies on participants to provide a fair and accurate description of an event they experienced.

The diary itself is a limitation, as to use it the participants must have access to a device that has web-access and can accurately present the diary. The design of the diary system must also be appealing and useable to ensure the participants use it, and do not encounter problems that discourage their use of the diary. The robustness of the diary and the fail-safe nature of the system were vitally important, as if the system went ‘offline’ or became unusable, there is no alternative data entry method, as there would be in the paper-based diary for example.

It was important that the diary study collected the right data from the participants at the right time, providing useable and useful data. If the questions asked during the diary were poorly formed, confusing, or simply asked the wrong question, it would significantly affect the quality of the collected data. Care was taken to check that the questions in the diary were grammatically correct, that the questions were clear, precise, and easily understandable, and presented the participants with single questions to respond.

While these limitations have the potential to curtail the collection of data with a diary study, steps were taken to minimise these risks. The participants were introduced to the diary, a ‘frequently asked questions’ section was added for immediate answers, and the diary was tested on multiple operating systems and devices, including the operating system and devices that the participants would be using to access the diary at their offices.

4.1.3 OBSERVATION

During the diary study the participants were also observed directly by the researcher. Observing events as they happen gives the researcher the opportunity to witness a phenomenon in real time and in the surrounding context. Observing events as they happen can lead to discovering unreported important factors, and provide a deep, immersed view of the environment surrounding the phenomenon.

Observation studies offer rich data, but also provide the researcher with the experience and knowledge of the environment in which the software engineers work. This contextual understanding aids in the future analysis of data and comprehension of responses in relation to other people or parts of the environment which the participants discuss. As an example it is difficult to understand what is meant by “shooting someone for asking a stupid question” until seeing a software engineer being shot from across the room with a “Nerf” gun if they ask a question perceived to be stupid by another software engineer in the room at the time.

The observation method was part of the method triangulation principle and provided an outside view of feedback received by software engineers, complementing their reported perceptions during the diary study. In addition to this, it provided vital experience in their environment that would serve to aid analysis and interpretation

OBSERVATION PROTOCOL

To ensure that the collected data was in a useable format, a protocol for recording notes, recording audio and taking photographs was produced. All notes were written down on a notepad, and include the date at the top of the page and a reference to the current location of the researcher at the time of taking the note. Notes taken were highlighted to signify where an instance of feedback was believed to have occurred, and then compared to the findings of the diary study.

The positional reference is an indication to the main observation locations used by the researcher, as can be seen in appendix B section 9.

Each note would include the time the individual note was taken, recorded in the left margin in-line with the note. Any quotations would be highlighted, and for any instance of feedback, a gold pen would mark the point and a blue sticky note would be added to the page. Audio of conversations between software engineers and other colleagues were recorded. Additionally, to ensure that the audio recordings taken during the observation could be related back to a time and place, a black square was drawn at the start and end of each audio recording on the observation notes, and assigned an incremental recording number.

In addition to audio recording and written notes, photographs were taken of the environment and of interesting features such as walls and pillars of quotations, as can be seen in Figure 4.1. The photographs captured the physical environment the software engineers worked in and the feedback that surrounded them in the form of quotations from customers, managers, and other members of both their team and other teams.

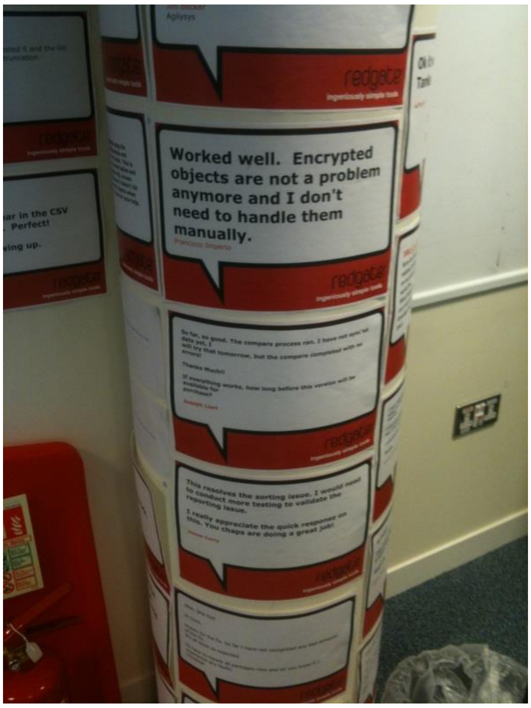
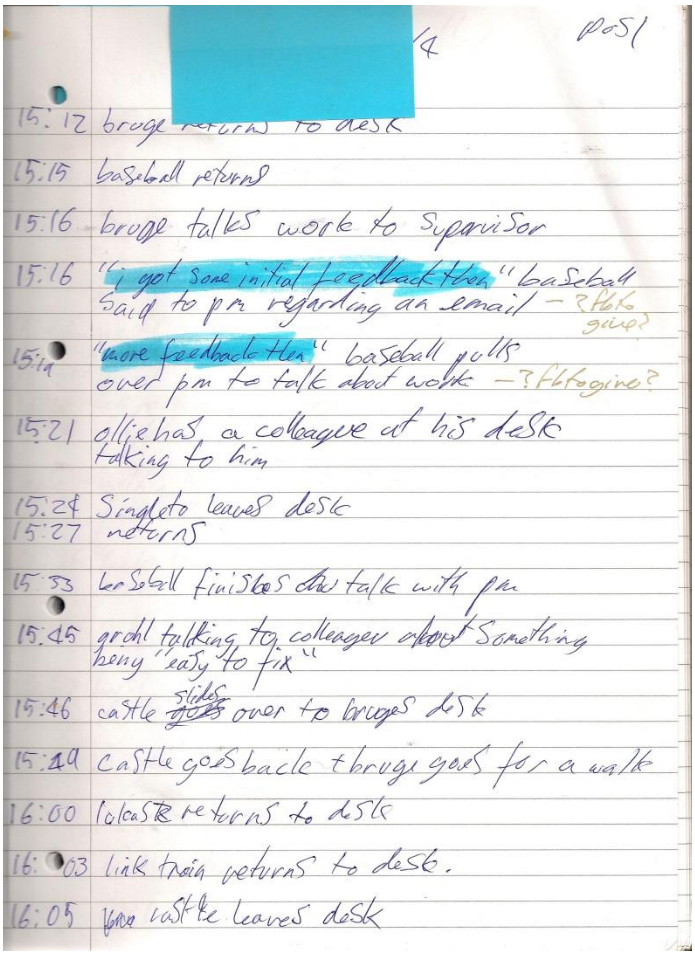

Worked well. Encrypted objects are not a problem anymore and I don't need to handle them manually.
So far, so good. The compare process ran. I have not sync'ed data yet, I will try that tomorrow, but the compare completed with no errors! Thanks Much!! If everything works, how long before this version will be available for purchase?
This resolves the sorting issue. I would need to conduct more testing to validate the reporting issue. I really appreciate the quick response on this. You chaps are doing a great job!
redgate
ingeniously simple tools

FIGURE 4.1 – PILLAR OF FEEDBACK

Any notes taken that were specific to individuals were done so using pseudonym names to ensure that the participants remained anonymous should the notes be mislaid. An encrypted file containing the reference and pseudonym names was retained in a secure, password-protected location. In line with the protocol observed during the interviews, all recordings were transferred from the original audio recording device and then deleted from the original device.

OBSERVATION LIMITATIONS

Observing events as they happened posed several limitations to the collection of data and the quality of the collected data. Access to the participants and their environment was limited, as although a seat near to the participants was acquired, it was not close enough to gain insight into what was actually happening, nor to provide an accurate account of any observed instances of feedback. As can be seen in Figure 4.2, the majority of the captured details relate to movements of the participants and distant fragments of conversations. These notes typically lack enough detail to fully express what was going on and what was being said.



[redacted] /4 PoS1

15:12 bruge return to desk
15:15 baseball returns
15:16 bruge talks work to Supervisor
15:16 "i got some initial feedback then" baseball said to pm regarding an email – ?fb to give?
15:19 "more feedback then" baseball pulls over pm to talk about work – ?fb to give?
15:21 ollie has a colleague at his desk talking to him
15:24 Singleto leaves desk
15:27 returns
15:53 baseball finishes the talk with pm
15:45 grohl talking to colleague about something being "easy to fix"
15:46 castle ~~goes~~ slides over to bruges desk
15:49 castle goes back + bruge goes for a walk
16:00 ~~lolcaste~~ returns to desk
16:03 link train returns to desk.
16:05 ~~Yoma~~ castle leaves desk

FIGURE 4.2 – OBSERVATION NOTES EXAMPLE

Access to the participants was negotiated through a manager, and the seating positions were typically close to teams, but not part of the actual teams. While this access provided a brilliant line of sight of the participants, the distance between the researcher and the participants, combined with the surrounding noise made it almost impossible to hear and understand any conversations that occurred, both as they happened and during playback of audio recordings.

Using an observation research method relies on the awareness and recording ability of the researcher, and the researcher's ability to identify and gain access to suitable areas from which to observe. Being unable to identify events that should be recorded, incorrectly or incompletely recording details of witnessed events, and failing to gain access to suitable observation areas can pose limitations to the quality of the collected data.

4.2 PERSONALITY INVENTORY

The participants were invited to complete a personality inventory. A personality inventory was included for two key reasons: (1) to provide a basis of comparison across the participants allowing the analysis to look at personality as a possible factor influencing the responses or actions of the participants, and (2) to indicate whether the participants were representative of the wider software engineering community. The latter this was done by comparing the results from the personality inventories of the participants of this study with personality data collected from other software engineers (mypersonality.org, 2012).

PERSONALITY INVENTORY MODEL SELECTION

Three models of personality were considered for their appropriateness and accessibility for this research: the Sixteen Personality Factors Model (16PF), the Myers-Briggs Type Indicator (MBTI), and the Five Factor Model.

The three personality models offer different interpretations of personality, and are not directly comparable. The Myers-Briggs Type Indicator is based on categorising individuals as one of sixteen different personality types by measuring individual preferences to four dichotomies. The Myers-Briggs Type Indicator is used widely, often referred to as the world's most popular personality inventory instrument (Myers, 1998). However, there has been some criticism of the MBTI instrument:

> *"There was no support for the view that the MBTI measures truly dichotomous preferences or qualitatively distinct types, instead, the instrument measures four relatively independent dimensions." (McCrae and Costa, 1989: 17)*

McCrae and Costa (1989) continue their criticism of the MBTI by arguing that the MBTI lacks independent evidence. Balijepally *et al.* (2006) argue that from their literature review, the Five Factor Model…:

> *"provides better measures for all factors that are measured by MBTI, it also allows us to assess Neuroticism" (Balijepally et al., 2006: 5).*

In contrast to the MBTI, the Five Factors Model has substantial evidence supporting its validity and reliability (Barrick and Mount, 1991). Evidence suggests that the Five Factors Model is generalizable across all cultures (McCrae & Costa, 1998; Salgado, 1997) and remains stable over time (Costa & McCrae, 1998). However, critics of the Five Factors Model claim that the model lacks scope (Paunonen, 2000), and that is the model is based only on empirical findings and has no underpinning theory, with Eysenck (1992: 671) reiterating an earlier statement that "no personality dimension can be taken seriously unless it is supported by theory" (Claridge, 1986).

The Sixteen Personality Factors model has also been subjected to criticism. Despite attempts, many other psychologists have been unable to replicate the findings presented by Cattell (1946). Howarth and Brown (1971) found 10 factors that did not relate to items present in the model, and findings by Sells *et al.* (1970) and Kline and Barrett (1983) could only verify 4 of the 16 factors. Cattell himself, in response to critics, published the results of his own factor analysis of the sixteen personality factor model, which itself failed to verify the 16 personality factors (Strelau and Eysenck, 1987). While there has been much criticism of the 16PF Model, the findings and Cattell's work laid the groundwork and led to investigation causing the discovery of the Five Factors Model by researchers including Fiske (1949), Norman (1963) and Tupes and Christal (1961).

A recent project at the Cambridge University Psychometrics Centre uses Facebook © to collect personality data (mypersonality.org, 2012). The mypersonality.org project uses questions from the International Personality Item Pool (IPIP) proxy for Costa and McCrae's NEO-PI-R domains, an instrument used to measure the Five Factors. Since the project began in 2008, they have collected personality data from over 3.6 million people around the globe. The mypersonality.org project has made much of its data available to fellow academics after an initial screening process, as well as removing any of the participants' personal details.

The evidence supporting the validity and reliability of the Five Factors Model and research indicating the appropriateness of online research using the Five Factors Model (Buchanan *et al.*, 2005) supports the use of the Five Factors Model in this research. Considering this, and the easily accessible recent data collected from software engineers (mypersonality.org, 2012) using an instrument based on the Five Factors Model, the Five Factors Model is the most appropriate personality model to use in this research.

PERSONALITY INVENTORY DESIGN

After deciding to use the Five Factors model of personality, it was important that an instrument used to measure these factors was chosen that made the collection of data efficient, did not require an excessive amount of time from the participants, and was comparable with other personality inventory data. To enable a strong, robust comparison with other data, this study used same the personality inventory questions as those used in the mypersonality.org project.

The mypersonality.org project uses the International Personality Item Pool (IPIP) proxy for Costa and McCrae's NEO-PI-R domains as the foundation for their personality inventory, including the questions, the scoring system, and the administration protocol, which are all in the public domain.

The personality inventory instrument was built as part of the diary study for all users who took part in the diary, and was it also made available to all other non-diary participants through a participant-specific URL. After an introduction to the personality inventory, the instrument asked each participant to rate 100 statements on a 5-point scale from very inaccurate to very accurate. A screenshot of the personality inventory instrument interface as used by the participants can be seen in appendix B section 8.

PERSONALITY INVENTORY ADMINISTRATION

The participants who agreed to participate in the diary study received an email containing details about the diary study before they began the diary study, which included information relating to the personality inventory. All other participants who did not wish to participate in the diary study received a different email which provided them with a unique URL and asked them to complete the personality inventory at their leisure.

The personality inventory scores were all stored in the same secure location as the collected diary study data, and with a unique anonymous reference for each participant in order to maintain confidentiality. For the diary participants, once they had completed the personality inventory, the link to the personality inventory which was found in the online diary study system was removed.

PERSONALITY INVENTORY DATA

In total, 12 diary participants and 5 non-diary participants completed the personality inventory, yielding 17 personality inventories collected from the original 24 interview participants.

PERSONALITY INVENTORY ANALYSIS DESIGN

The analysis of the data collected in the personality inventory is paired with a standard analysis design. A method of interpreting each individual's response is included with the personality inventory questions, indicating how each response should be calculated to identify an individual's personality score in each of the five personality facets interpreted by the Five Factor Model.

PERSONALITY INVENTORY LIMITATIONS

The number of responses limits the usefulness of the personality inventory. It was not possible to identify statistically significant correlations at the individual respondent level due to the small number of responses to the personality inventory.

Similar to the web-based electronic diary, the participants required access to a device that had internet access to complete the personality inventory. The stigma attached to personality inventories may have reduced the response rate among the participants, and may explain why 7 participants did not complete the personality inventory. As the personality inventory was not used to explain behaviour or individual responses to questions, the limitation of the instrument and measuring technique itself do not affect the use of the instrument during this research.

4.3 DATA ANALYSIS DESIGN

The four research methods each required a different analysis design.

Semi-structured Interviews. The interviews provided rich data. An inductive analysis identified emergent themes using a "multi-pass" approach of verbatim-gist-superordinate as discussed by Rugg and Petre (2004: 156-158). After counting the exact words or phrases used by the participants (verbatim analysis), the "synonymous alternative phrasings" (ibid: 157) of the respondents' words and phrases were identified (gist analysis), and finally the emergent themes were found by identifying the overarching meaning of the participants' responses and grouping responses together (superordinate analysis).

Diary Study. The diary study combined the collection of rich data with the collection of specific, pre-listed selection data. The data was analysed to identify the choices the participants made using a basic count. The rich data was analysed using the verbatim-gist-superordinate approach as discussed above.

Observation. An inductive analysis of the written notes was planned and piloted, but provided incomplete findings. The observation research method provided useful contextual information about Red Gate and improved the researchers understanding of the environment and the participants.

Personality Inventory. The personality inventory came with a set analysis approach. The set analysis calculates an individual's score in each of the five factors from the individual's responses to each question. Each question affects the individuals score in one of the five factors, and each questions has a positive or negative modifier. The participant's responses to each question are scored from 0 to 5. The participant's accumulative score for each of the five factors is then divided by 20, and their final score in the range of 0 to 5 for each of the five factors is identified.

The analysis design for the semi-structured interviews, the diary study and the observation research methods were piloted (section 5.1), and examples of the analysis are presented in sections 5.4.1 (semi-structured interviews) and 5.4.2 (diary study). The personality inventory analysis was not piloted as the procedure was provided as an exact calculation of the participants responses to each question.

4.4 DATA SUMMARY

In total, the research completed in the Feedback in One Software Engineering Environment Study (described further in chapter 5) generated over 15 hours of recorded audio from interviews, over 200 pages of written notes from observation and interview notes, 76 recorded instances of received feedback, 45 completed day summaries, and over 130 photographs of the environment which the software engineers worked.

4.5 ETHICAL CLEARANCE

To ensure that the research was appropriate and posed minimal risk to participants, ethical approval was obtained from the Open University's Human Research Ethics

Committee (appendix B section 3). A copy of the consent form presented to the participants is also included in appendix B section 4.

4.6 RESEARCH DESIGN SUMMARY

The use of method triangulation to investigate a phenomenon supports the validity and the reliability of the findings, and underpins the discovery and identification of any important factors surrounding the phenomenon.

The interviews provided reflective, “from-memory” accounts of what motivates and de-motivates software engineers, how they received general feedback and how it made them feel. The diary study provided specific, detailed, near-real-time participant reported instances of feedback and how it affected them, as well as rich descriptions of how feedback affected how they felt at the end of a working day.

The observations enabled the researcher to gain first-hand experience of the participants’ environment, greatly aiding the understanding and analysis of the participants’ data. While feedback was witnessed during the observations, the collected observation data was insufficient to be useful in identifying the specific details of feedback that the participants received. The personality inventory enabled the comparison of the participants to the wider community of software engineers.

The reliability of the data was increased by investigating what software engineers thought about the impact of received feedback. Feedback was investigated through reports of instances of feedback, through day-specific reflections of feedback impact, and by questioning what software engineers believed feedback was, how it was received, and how it affected them. This provides a useful comparison between what the participants recalled as being their reaction to received feedback weeks or months after receiving it, and what they reported as their reaction to received feedback within minutes of receiving it. It is possible that during the interview they could have reported what they *think* they experienced, and not what they actually experienced. The diary study and the near-to real-time data collection reduced the potential for the participant’s memory to distort the reported instances of feedback.

CHAPTER 5

FEEDBACK IN ONE SOFTWARE ENGINEERING ENVIRONMENT

This chapter presents the findings from a study investigating feedback in software engineering. The research approach and research methods used during this study were described in chapter 4. Data were collected from multiple teams working at Red Gate, a software development company based in Cambridge, UK.

This study builds on the findings of the Feedback Scoping Study (chapter 3) by investigating feedback in one software engineering environment and extending the identification of feedback characteristics to identify different forms of feedback found in software engineering, and investigate how different aspects of software engineers can be affect by received feedback.

The following sections discuss the completed pilot study (5.1), present a detailed description of the Red Gate environment and the different divisions and teams in each environment (5.2), present the participants (5.3), present the data collected during this study (5.4), identify the results of the data collection and analysis (5.5), discuss the threats to validity (5.6), summarise the collected data (5.7), summarise findings and identify emergent questions (5.8), and finally present a state of play updated with the knowledge gained from the findings of this study (5.9).

5.1 PILOT STUDY

To validate the protocol, check that the method of investigation was suitable, pilot the data analysis, and to ensure that the terminology used was appropriate, a pilot study of the proposed research design was completed. To pilot the interview and diary study, five pilot participants were contacted and subsequently agreed to participate. All of the participants identified themselves as software engineers with experience working in software development environments.

All of the participants completed the interview either directly in person, or via the internet using Skype. The participants were asked post-interview to provide any comments they had on the interview, the questions asked, and the terminology used. Their responses to each question were evaluated to ensure that the participants had interpreted the question as it was intended to be, and that the collected data would be appropriate and useful for the study's intended investigation. The intended analysis approach (4.3) was piloted, identifying themes emerging from the participants responses, and checking that the protocol for the analysis was appropriate.

The five participants were asked to complete the online diary for one day only, and provide comments on the system and their experience using it. The participants were asked to comment on the design and usability of the diary, the questions being asked of them, and their understanding of and ease in answering each question. Their responses were evaluated and their comments recorded in order to assess whether the online diary was suitable for the intended investigation.

The five pilot participants worked in different locations across the UK, and it was not practical to gain access to them or another software development team environment to pilot the observation method. Instead, researchers at the Open University were observed. The collected data from teams at the Open University was analysed to pilot the observation analysis design. The pilot of the analysis identified that more information was required to be recorded in the written notes to ensure that the audio recordings could be related to the written notes.

In addition to the above identified pilot studies, knowledge gained during initial meetings with Red Gate personnel was used to adapt the terminology to be more appropriate to the participants, and more likely to relate to their work and their current working environment.

5.1.1 PILOT STUDY OUTCOMES

After completing the pilot study with the five participants and analysing their responses and comments for each research method, only minor modifications were required. The pilot studies indicated that the research design would yield appropriate data for the intended investigation, and the required changes were mainly to aid understanding, to improve clarity, and to make the terminology more appropriate for the environment of the intended participants. The knowledge of

Red Gate's agile, scrum-based development environment shaped the questions to address stand-up meetings and sprint reviews, as well as referring to team leaders as line managers, and understanding the division-based structure used at Red Gate.

For the interview, the changes required included modifying the questions to use the term 'project' instead of 'work', referring to the participants as 'software engineers', minor separation of questions that were incorrectly worded to ask two questions or lead the respondent, and to include a check list of discussion points that were implicitly used in the pilot study without being explicitly included during the interview method design. This list of discussion points was used to ensure that all the key areas of the investigation were covered and to ensure that the participant's opinion on a range of topics was investigated, if it had not come up during open-ended questions during the interview (appendix B section 1).

For the diary study similar changes were required. The terminology used by the participants, such as stand-up meetings, projects, project manager, and division head all needed to be included in the diary in place of other terms such as work, meetings, manager, and senior manager. These changes did not alter the design of the diary study, but aided the participants' understanding of the questions and made the questions appropriate for their current working environment.

By piloting the observation method it was possible to identify and establish that the protocol in place to collect notes was efficient and appropriate for the intended investigation. Notes would be taken with a reference to the time, the location, and the participants being observed. No required modification to the observation protocol was identified during the pilot study.

5.2 STUDY ENVIRONMENT

The design of this research study required an environment that included four key characteristics:

1. A team-based environment.
2. Access to software engineers.
3. Access to multiple teams of software engineers.
4. Willingness to participate by the software engineers.

Red Gate was identified as a suitable company which had already expressed an interest in studies investigating the motivation of software engineers. It offered access to multiple teams, and appeared interested in the research. After a visit to their offices, it was apparent Red Gate cared about feedback, as it was on display on almost every wall in all of the different software development team environments. Red Gate was both a pragmatic study environment and an ideal study environment, as it offered access to software engineers who were willing to participate, and from the environment it appeared that Red Gate had an interest in providing and encouraging feedback.

The Red Gate environment is presented in more detail in the following sections, including specific details about the different divisions and the different teams at the time of the study. For photos of the environment please refer to appendix B section 9. Observational data was collected from four of the five different divisions, as identified by the reference of positions A-E as shown in figures illustrating the layout of the office environments at Red Gate, which can be found in appendix B section 9.

5.2.1 RED GATE ENVIRONMENT

Red Gate is a software development company based in Cambridge, UK that employs over 200 people, including approximately 30 software engineers. Red Gate mainly specializes in SQL tools for database administrators, but with one division entitled “new business”, it suggests that Red Gate is not focused solely on SQL tools.

The software engineers work in small teams, ranging in size from two to ten people, with the number of software engineers in each team between one and five. Other members of the teams include software testers, a project manager, shared technical authors, and shared user experience experts. All of the teams work in an agile scrum-based development environment, but each team has its own implementation of the scrum method.

Red Gate was founded in 1999, and had expanded over the recent years with continuous growth from a two-person organisation to one that, as of 2010, employed over 200 people spread across two floors of a shared building. Its focus was to attract creative, intelligent people, and to do this they offered an attractive salary (starting salary £30k for software engineers) and benefits package, including BUPA health care, discounted gym membership, flexi-time, high-spec equipment, free hot good-quality breakfast and lunch, snacks, and a 5-week sabbatical after 5 years with the company.

The software engineers worked in small teams, based in large open-plan offices. Every software engineer had their own large desk with a minimum of two monitors, a high-spec computer, and reportedly almost any work-focused office or electrical supply they request. While the software engineers were all provided with company chairs, they were all allowed to make modifications, with one engineer seen sitting on a soft, bouncy ball.

Further amenities provided to the employees on-site included a foosball table, a table-tennis table, a subsidised vending machine, a video library, a book library, and a hammock. During the study, it was common to see these in use by the employees. Often development teams, including software engineers and software testers, would stop working and go play foosball or table tennis together at seemingly unscheduled times.

The Red Gate environment tries to nurture great ideas and quality work from its staff with the environment, but it also provides performance-based rewards. In one area of the company, there was a big screen that displayed the current monthly sales, and their current monthly sales target. Any month when Red Gate met their sales target, all of the employees received £1,000 in their account the next day, and at a later date they were given an additional ‘reward’. One previous reward was a special visit to the Apple store with a £300 voucher each, and the most recent reward reported during the study was a closed-store visit to John Lewis and a £500 voucher for each Red Gate employee. The rewards were reportedly given to all employees who had been at Red Gate for at least three months prior to Red Gate meeting the target.

Each division also arranged team and division events, such as trips away and meals out. During the study an entire division finished work early and went out for an event, a meal paid for by Red Gate. It was reported that previously an entire development team was sent away for a week to a large country house to work on solving a problem together, with the results of their week away instigating a new project for Red Gate.

All of this suggests that Red Gate is committed to hiring productive software engineers by offering an attractive salary and benefits package, providing them with an enhanced environment and office space designed to help them to work, removing basic needs such as hunger and thirst, and encouraging team cohesion and active team-building with foosball, table tennis, and events such as meals as a team and/or division. In the following section more detail is provided on each of the Red Gate divisions.

5.2.2 RED GATE DIVISIONS

Red Gate was split into five different divisions. These divisions each have a division head, and contain one or multiple development teams. There are some employees who work across divisions, and there are some employees who work across teams. Typically, each software engineer is part of one team, but may work on projects part time with other teams. The different divisions are listed below:

1. **SQL** – This division has four teams who all work on Red Gate's 'flagship' SQL tool products.
2. **Database** – This division has two teams who work on developing and maintain the database products.
3. **New Business** – This division has one 'team', but is more like a collection of people all working on different and similar products at the same time. All of the people in this division are investigating new avenues and new products for Red Gate to potentially develop.
4. **DevOps** – This division has one team responsible for maintaining, troubleshooting, and developing all of Red Gate's internal systems used by the other employees. Typically, they spend their time helping other people to use the Red Gate systems or solve a problem, in addition to developing new systems.
5. **.NET** – This division has one team working on the .NET products.

All of the software development teams at Red Gate work in an agile environment, using a variation of scrum (Schwaver and Beedle, 2002). The level of implementation of scrum varied greatly, with some teams having daily stand-up meetings and defined sprints, and other teams appearing to never have a daily stand-up meeting or a defined sprint. The hierarchy of the teams varied in line with the scrum implementation. The teams with stand-up meetings had a clear and identifiable project manager[3], whereas the teams without stand up meetings did not have a clearly identifiable project manager.

From the observation it appeared that in all of the divisions the teams communicated freely, discussing both work and personal interests openly and

[3] Used by the participants and Red Gate management to represent the scrum master.

often. The conversations would typically be work related, technically related, or social discussions. Some teams appeared to have social discussions in the majority of their conversations, and others appeared to never talk socially. Typically the teams for any one division would all sit in the same area (Figure 5.1). The two CEO's of the company had roaming desks in various locations so they could spend time in each division whenever they wanted.

SQL DIVISION ENVIRONMENT – POSITION A & E

Located on the first floor of the building, the development teams in the SQL division took up over half of a large open-plan office, which they shared with marketing and support. Figure 5.1 provides the layout of the desks for the SQL team, with the highlighted areas indicating the four different teams. The image also indicates the first observation location (position A) and the fifth observation location (position E), as well as providing pseudonym names for all of the participant software engineers.

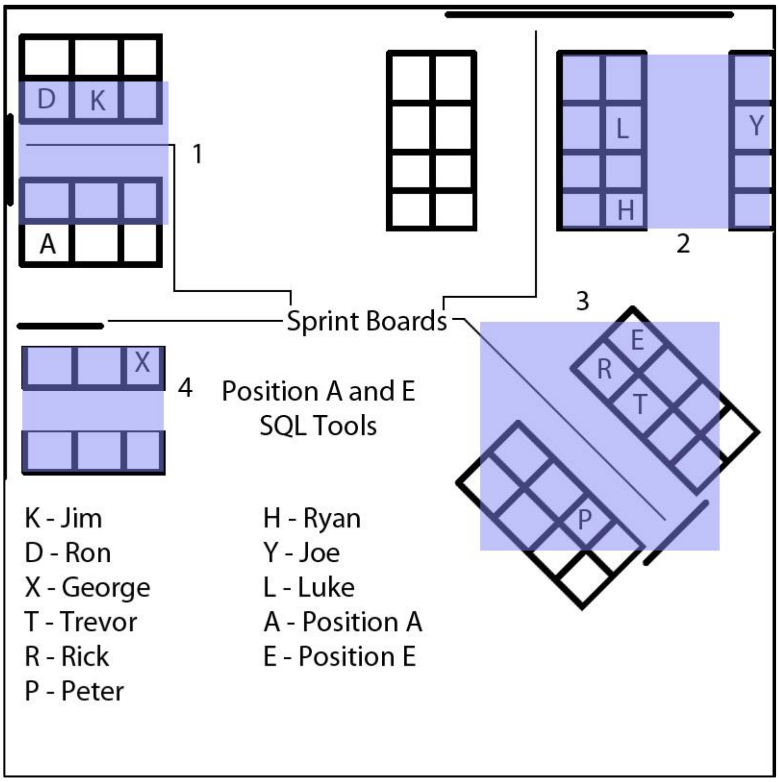


Figure 5.1 – SQL Team Layout

Desk locations without a letter either mean an empty desk, or more likely a desk occupied by a member of the team who did not participate in this research, such as a project manager, software tester, user experience expert, or technical author. The numbering of the teams is of no relation to any number system used at Red Gate.

The teams in the SQL division all used sprint boards, and had daily stand-up meetings where they each took it in turns to discuss their progress. All members of team participated in the stand-up meetings, each discussing their individual progress, led by the project manager.

TEAM DEMOGRAPHICS

There were 4 main teams in the SQL division. Team 1 consisted of 2 software engineers, 1 software tester, 1 project manager, and a shared user experience expert. Team 2, the largest team in the SQL division, ranged between 7-10 members, bringing in members from other teams as required. Team 2 typically had 3-4 software engineers, a project manager, 1-2 software testers, a user experience expert, and a technical author.

Team 3 is the second largest team in the SQL division, and had 3 software engineers, 1-2 software testers, a project manager, a shared user experience expert, and a shared technical author. Team 4, the smallest team in the SQL division, had 1 software engineer, 1 project manager, 1 software tester, a shared user experience expert and a shared technical author.

DATABASE DIVISION ENVIRONMENT – POSITION B

The database division included 1 development team and was located in the corner of a large open-plan office on the first floor of the building, sitting in close proximity to the head of software development and other heads of different departments. Figure 5.2 shows a plan of part of the open-plan office area, including the work space for the development team, which consists of 8 chairs in a back-to-back layout.

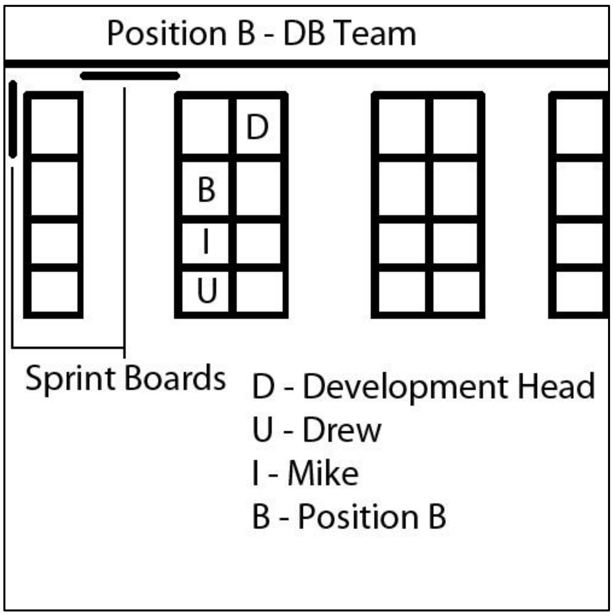


FIGURE 5.2 – DATABASE TEAM LAYOUT

The team consisted of approximately 5 software engineers, including 1 software engineer, who was also the team's project manager, and 3 software testers, all working on multiple projects at the same time. The location of the 2 participants in this study, named Mike and Drew, are shown in Figure 5.2.

The team had a stand-up meeting each day, but not all members of the team participated. The meeting was typically led by the project manager, who asked questions of the other members present in the stand-up meeting. The team used a sprint board to track their progress, with members taking individual sticky-notes and sticking them to their screens while working on a particular task.

NEW BUSINESS DIVISION ENVIRONMENT – POSITION C

The new business division was located on the second floor of the building, and consisted of 1 team. The team included software engineers, user experience experts, and experienced individuals in software-related fields. While they are identified as a team, they all worked at various stages on individual and team projects, with the *modus operandi* of "fail fast and fail often". The layout of the new business environment is shown in Figure 5.3.

The new business team was spread out over half of an open-plan office, with team members in some cases sitting the other side of the room. The team had roughly ten members, and approximately six of them regarded themselves as being software engineers, however the participants expressed that their job is different to that of the other software engineers at Red Gate, and is often different day to day.

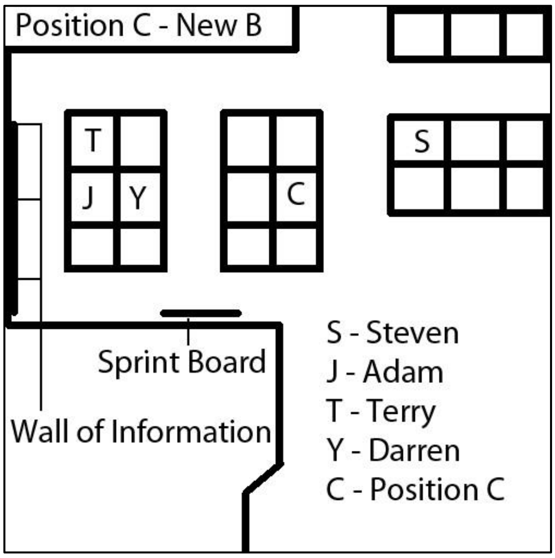


FIGURE 5.3 – NEW BUSINESS TEAM LAYOUT

The team used a sprint board and a window of information to track their progress, and typically do not have a stand up meeting each day. The team members all had different versions of 'nerf' guns, which they used to shoot across the room at each other for various excuses, such as finding a bug in someone's code, or if someone asked a question which is considered to be 'stupid'.

DEVOPS DIVISION ENVIRONMENT – POSITION D

The 'Devops' division had 1 team, and they were located along a stretch of an open-plan office on the second floor. The Devops team had roughly 10 members, and included 5 software engineers. As the team worked on maintaining, supporting and developing the in-house software used at Red Gate, the duties and tasks of a typical day varied, from days where they spent all of their time developing new software, to days where they spent all of their time helping their colleagues, and days with a mixture of both of those tasks.

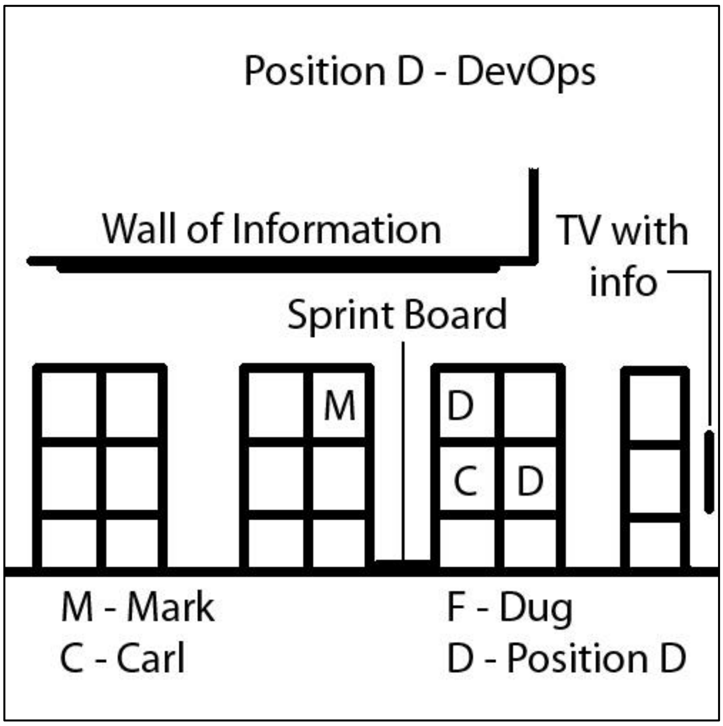


FIGURE 5.4 – DEVOPS TEAM LAYOUT

The Devops team used a TV with information on to keep track of any bugs they need to fix, and to keep up to date with the build information of their systems. They all sat together with back to back desks (Figure 5.4). The team used a sprint board, and they had a large wall with lots of information, ranging from relaxed jokes, to specific user and browser information.

The devops team had stand-up meetings, but they were unscheduled and did not happen daily. When the stand-up meetings did happen, they were often with varying members of the team, and did not include everyone. Typically a stand-up meeting would include two to three members of the team, and was specific to a project they were working on, rather than the entire sprint board.

.NET DIVISION ENVIRONMENT

The .NET division was a small division located on the second floor of the building. The division had 1 team, which consisted of 3 software engineers and one software tester. As the team were not part of the observation stage, a layout of their environment and a position for the observation is not included. The team all sat together, and had daily stand-up meetings. The team also used a sprint board. The team sat in close proximity to some of the Red Gate support staff, and in the same open-plan office as the Red Gate human resources department.

5.2.3 FEEDBACK ON DISPLAY IN THE RED GATE ENVIRONMENT

There were varying levels of feedback displayed in each Red Gate division environment. Some divisions had no obvious feedback related to their work, and other teams had feedback stuck to the front of the desk divider in their individual space. To ensure clarity and to be precise on the changes between divisions relating to the presence of feedback, the following sections identify and discuss the feedback on display in each separate Red Gate division.

FEEDBACK IN THE SQL DIVISION ENVIRONMENT

The SQL division environment is littered with examples of feedback, as shown in Figure 5.5 and Figure 4.1. The division head, project managers and software engineer manager have ensured that the environment their teams work in included substantial amounts of feedback from a range of different sources, including fellow software engineers, members of other development teams, other employees, and customers of Red Gate.

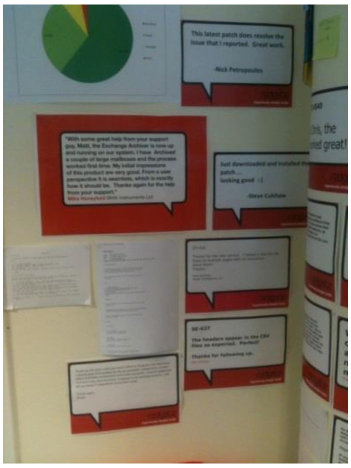

This latest patch does resolve the issue that I reported. Great work.
-Nick Petropoulos
"With some great help from your support guy, Matt, the Exchange Archiver is now up and running on our system. I have Archived a couple of large mailboxes and the process worked first time. My initial impressions of this product are very good. From a user perspective it is seamless, which is exactly how it should be. Thanks again for the help from your support."
Just downloaded and installed the patch ... looking good :-)
-Steve Culshaw
The headers appear in the CSV files as expected. Perfect!
Thanks for following up.

FIGURE 5.5 – FEEDBACK ON A WALL

FEEDBACK IN THE DATABASE ENVIRONMENT

There was no obvious feedback on display in the database environment. All of the walls contained details specific to projects, and all of the developers' working environments only contained their computers and relevant books. There is not a lot of space to display feedback, so it is possible the lack of feedback compared to other divisions is due to the special restrictions rather than by choice.

FEEDBACK IN THE NEW BUSINESS ENVIRONMENT

Feedback was displayed on walls and pillars around the new business environment. Due to the relation of the walls and pillars, the feedback was not always clearly visible to the team, and appeared to be a lower priority when allocating space in the environment. A lot of the space on the walls was used for more project-focused information, such as details of what needed doing and what problems they had, or more humorous friendly topics, such as images of one of the software engineers holding a ‘nerf’ gun below the headline “I refuse to negotiate with software terrorists”.

FEEDBACK IN THE DEVOPS ENVIRONMENT

There was a lot of feedback on display in the Devops environment. As the team’s main and only ‘customers’ are the other members of Red Gate, it would be a lot easier for them to gather feedback, and this is shown in their environment. They had an entire wall filled with a mixture of feedback, project tasks, and humorous banter, this can be seen in Figure 5.6 and Figure 5.7.

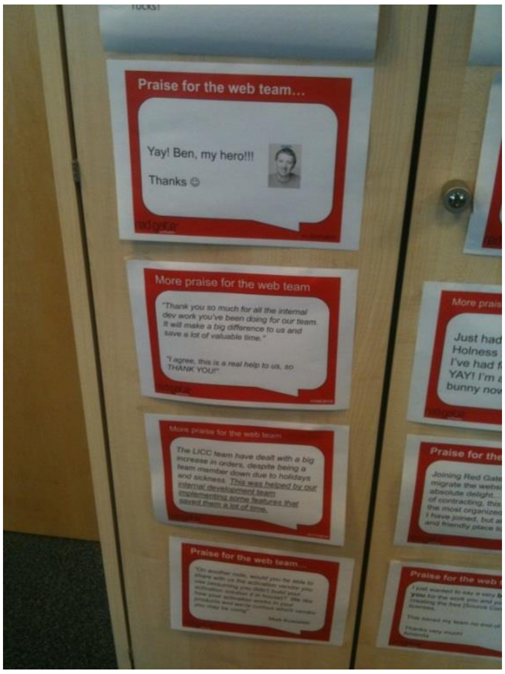

Praise for the web team...
Yay! Ben, my hero!!!
Thanks ☺
More praise for the web team
More praise for the web team
Praise for the web team...

FIGURE 5.6 – FEEDBACK IN DEVOPS

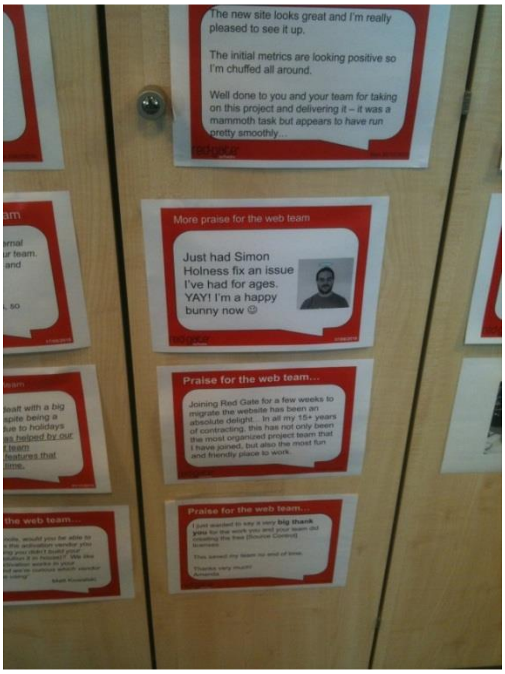

The new site looks great and I'm really pleased to see it up.
The initial metrics are looking positive so I'm chuffed all around.
More praise for the web team
Just had Simon Holness fix an issue I've had for ages. YAY! I'm a happy bunny now ☺
Praise for the web team...
Praise for the web team...

FIGURE 5.7 – FEEDBACK IN DEVOPS

The feedback ranged from generic thank you comments, to really detailed explanations of the problem someone experienced and the help they received from either one person or the entire Devops team. As the wall is within close proximity of every member of the Devops team, they were able to see it and walked past it every day.

5.3 PARTICIPANTS

In total there were 24 participants. All of the participants were male, and all of them reported having a university degree. The participants all had a minimum of 6 months experience as a software engineer, with the most experienced software engineer reporting 24 years of industry experience. The participants had been employees at Red Gate for between 6 weeks and 6 years.

The first group of participants consisted of 24 engineers who agreed to participate in the interview study. After the interview the participants were asked if they were willing to participate in the diary study, and subsequently 15 participants completed the diary study. There are 17 responses to the personality inventory, collected online through either a unique URL, or as part of the participant's diary study. Table 5.1 shows a summary of the participants and dates of data collection for each research method.

TABLE 5.1 – RESEARCH PARTICIPANTS

	# of Participants	Data Collection Dates
Interview (I):	24	28th March 2011 – 8th April 2011
Diary Study (DS):	15 (of the 24 from I)	11th April 2011 – 15th April 2011
Observation Study (OS):	15 (same as the DS)	11th April 2011 – 15th April 2011
Personality Inventory (PI):	17 (of the 24 from I, 12 of the DS)	11th April 2011 – 20th April 2011

Table 5.2 presents the details of the software engineers' years of experience, how long they had been employed by Red Gate, the software development environments in which they had development experience, and their level of formal education. Also included in this table are details specific to each software engineer's team, such as team size, and how many other software engineers were in their team. The final three columns indicate each software engineer's participation in the interview (I), personality inventory (PI), and diary study (DS).

TABLE 5.2 – PARTICIPANT DEMOGRAPHICS

#	Current Role	Experience	Time at Red Gate	Environment	Education	Engineers in team	Team size	I	PI	DS
P1	Engineer	10 years	5 years	scrum	Computer Science Degree	2	6	Yes	Yes	No
P2	Engineer	3.5 years	3.5 years	scrum	University Degree	2 to 3	4 to 5	Yes	Yes	Yes
P3	Engineer 75% Project Manager 25%	2.5 years	2.5 years	scrum	Computer Science Degree	2	3	Yes	Yes	No
P6	Engineer	10 years	9 months	scrum, Waterfall	University Degree	5 to 6	3	Yes	No	Yes
P7	Engineer	20+ years	6 weeks	Agile, range of others	University Degree and Masters	5 to 6	5 to 6	Yes	Yes	Yes
P9	Engineer (1.3 years)	1.3 as SE	3.3 years (1.3 as SE)	scrum	Psychology Degree	1 to 4	4 to 10	Yes	Yes	No
P10	Engineer	5 years	5 years	Agile	Computer Science Degree	2	2 to 12	Yes	Yes	No
P12	Engineer	14 years	4 years	Agile, Waterfall	Information Engineering Degree	5 to 6	5 to 6	Yes	No	No
P13	Engineer	22 years	3.5 years	Agile	University Degree and Masters	5 to 10	5 to 10	Yes	No	No
P14	Engineer 20% Project Manager 80%	10 years	6.5 years	Agile	Bio-chemistry Degree and Masters	3 to 4	6 to 7	Yes	No	Yes
P15	Engineer	6 months	6 months	Agile	Computer Science Degree	1	1	Yes	Yes	Yes
P16	Engineer	16 years	8 months	Waterfall and Agile	University Degree	1 to 6	1 to 6	Yes	Yes	Yes
P17	Engineer	24 years	5 years	Test Driven	University Degree	10 to 11	10 to 11	Yes	Yes	No

#	Current Role	Experience	Time at Red Gate	Environment	Education	Engineers in team	Team size	I	PI	DS
P19	Engineer	6 months	6 months	Test Driven	University Degree	8	8 to 13	Yes	Yes	Yes
P20	Engineer	10 years	1.5 years	Agile	Computer Science Degree	7	7	Yes	Yes	Yes
P21	Engineer	4 years	1-2 years approx.	Agile, range of others	Computer Science Degree and Masters	2	5	Yes	Yes	Yes
P22	Engineer	9 years	1-2 years approx.	Agile	Computer Science Degree	2 to 4	5	Yes	Yes	Yes
P23	Engineer	3 years	9 months	scrum	Computer Science Degree	3	7	Yes	No	No
P24	Engineer	2 years	1 year	scrum	Computer Science Degree	6 to 7	6 to 7	Yes	Yes	Yes
P25	Engineer	1.5 years	1.5 years	scrum	University Degree	3	10 to 11	Yes	Yes	Yes
P26	Engineer	5 years	1 year	Agile/scrum	Computer Systems Degree	1	2	Yes	Yes	Yes
P27	Engineer 50% Project Manager 50%	10 years	1 year 2 months	scrum	University Degree	2.5	5	Yes	No	Yes
P28	Engineer	4 years	9 months	Agile, Waterfall	Computer Science Degree	8	8	Yes	Yes	Yes
P29	Engineer	3 years 8 months	8 months	scrum	Computer Science Degree and Masters	2	4	Yes	No	No

KEY:

- I – Interview Participation
- PI – Personality Inventory Participation
- DS – Diary Study Participation

5.4 RESEARCH DATA

The data collected in this study was collected from software engineers working at Red Gate over four weeks in March and April 2010. The collection of interview data spanned over two weeks, followed by the collection of personality inventory data and the completion of the simultaneous observation and diary study, lasting one week. In the following section each step of data collection is identified, and an overall view of all the data collected from this study is presented.

DATA MANAGEMENT

The first step in the data preparation was ensuring the complete confidentiality of the participants. In accordance with the consent form the participants signed, all of their data was stored securely and their confidentiality maintained.

Each of the participants was assigned a unique participant number (Table 5.2) as a reference for all of their data. This ensured the confidentiality of the participants, while enabling their data to be analysed across multiple research methods using their participant number as a reference. In addition to the participant numbers, each participant was also assigned a pseudonym (Table 5.2).

To ensure there was no possibility of losing the participant number to participant relationship, a file containing the relationships was created, encrypted, and stored on a password-protected computer.

The data was collected in several forms across the multiple research methods. The interviews provided audio recordings, written notes and signed consent forms. The diary study and personality inventory provided electronic encoded collections of data related to participants. The observations provided written notes of what was seen and heard and audio recordings of conversations. It was important that all of this information was stored securely.

All audio recordings were transferred from the audio recording device, renamed with the participant’s unique name, and stored in a secure password-protected folder on a password-protected computer. All written notes were stored in a locked filing cabinet. All electronic data collected during the diary study and personality inventory was downloaded and stored in a secure password-protected folder on a password-protected computer.

To protect against losing such valuable data, a backup plan was prepared and followed. The audio recordings were backed up to two separate hard drives, both on password-protected computers. The electronic data collected during the diary study and personality inventory was also backed up to two separate hard drives, both on password-protected computers. Regular backups of all of the data were conducted, and basic integrity testing of the backups was completed.

Table 5.3 summarises the data preparation required for all the collected data. Different data preparation steps were required for the different types of data collected that ensured all data was stored in a useable format that maintained the confidentiality of the participants and the relationship to each relevant participant to enable data analysis.

TABLE 5.3 – DATA PREPARATION REQUIRED

Data Collected	Preparation Required
Consent Forms:	Electronic note of existence, stored in filing cabinet
Interview – Written Notes:	Shredded, analysis not intended
Interview – Audio Recordings:	Imported into NVIVO and transcribed
Diary – Electronic Data:	Online pages created to facilitate analysis
Direct Observation – Written Notes:	Stored in filing cabinet
Direct Observation – Audio Recordings:	Imported into NVIVO and transcribed
Personality Inventory – Electronic Data:	Online pages created to facilitate analysis

5.4.1 SEMI-STRUCTURED INTERVIEWS

The semi-structured interviews were designed and completed as outlined in the research design (chapter 4). In total, the interviews provided over 15 hours of audio, with each interview lasting between 18 and 80 minutes. The interviews were conducted in private rooms, with only the interviewer and interviewee present. All of the participants read, agreed and signed a consent form before the interviews began, and also agreed to the interviews being audio recorded to aid data analysis.

At the conclusion of each interview the participant was asked if he was willing to complete a further stage of research and participate in a diary study starting within two weeks' time. Of the 24 interview participants, 18 indicated that they would be willing to participate.

DATA PREPARATION

The audio recordings collected during the interviews were imported into NVIVO, where they were transcribed verbatim. Each segment of transcription contained the name of the person who was talking, what was said, and a time stamp at the start and end of the segment. It should be noted that transcription did not begin until all data was collected to avoid biasing any further data collection. The demographic information was extracted from the transcriptions and tabularised with all of the participants to get an overview of the demographics of all participants.

DATA ANALYSIS

The interview instrument and transcripts divide into three different topics: demographic information, motivation and de-motivation questions, and feedback questions.

An initial analysis approach used the demographic information combined with the participants' responses to the other two themes during the interview to investigate any relationship between responses and demographic information.

The second topic of the interviews focused on identifying factors that the participants reported as being motivating or de-motivating. The analysis was conducted question-by-question. The analysis per question was inductive, identifying emergent themes related to (de)motivation strictly on the basis of what was evident in the data, as presented in section 4.3. These themes were then extracted for each participant and analysed across the group of participants to identify any themes emerging from multiple participants. An example of the analysis process is discussed in the next paragraph.

In one question "Is there something that really saps your energy at work?" a participant responded, partly, with "[not] being able to [get] dedicated time, uninterrupted time". This was initially classified verbatim as *being interrupted*. At the next stage of analysis, the gist analysis classified this as *interruptions*, as the participant reported how he considered only checking his emails at two points during a day so that "emails can't interrupt" him, and stop him working. Finally, at the superordinate stage of the analysis, this was classified with other similar responses as the theme *obstacles*. The participant continued his response and discussed how people used headphones to avoid being interrupted, and how he enjoyed uninterruptable time and wished to avoid being interrupted so that he could "get [his] head into" his work.

The final topic of the interviews asked the participants about their experiences of feedback in their software development environment, and the impact the received feedback had on them. The analysis approach used was identical to the approach used to analyse the second topic questions, using a question-by-question inductive analysis to identify emergent themes strictly on the basis of what was evident in the data. The emergent themes were extracted and compared to all the participants in order to identify any themes emerging from multiple participants.

5.4.2 DIARY STUDY

Of the 24 interview participants, 15 completed the diary study. The diary study lasted 5 days, but some participants only completed 4 days due to absence. From the diary study, 76 instances of feedback and 45 completed day summaries were collected, with 6 participants completing a full week, defined as submitting 5 day summaries, or 4 if they were absent for 1 of the study days.

DATA PREPARATION

The diary study was completed online, and the data stored online. To aid future analysis, web pages were produced to connect to the database where the data was stored and display the user-submitted data in an accessible format for analysis. This included writing pages that presented the data for each individual, and pages that presented the data for all of the participants. Collected data was also exported and stored in a secure password-protected EXCEL workbook to aid analysis.

DATA ANALYSIS

The diary study collected online electronic data. The data collected was either instances of experienced feedback as reported by the participants or an end of day form on how the feedback they had received that day had affected how they felt at the end of the day. Initially, a numerically-focused approach was taken to identify the characteristics of the collected data, such as the number of instances of feedback submitted, the number of day summaries completed, and the type of feedback identified by the participants. Next, a more detailed analysis separating the collected feedback instances and day summaries data was required, as described in the following sections.

Feedback Instances. Each instance of feedback was extracted from the online database and formatted into a readable useable format. Each instance was analysed individually and notes were made on each instance. The feedback characteristics values reported in the feedback instances were investigated, and the coverage of different combinations of values for the feedback characteristics in the instances was evaluated. Similar instances of feedback were identified by comparing the reported values of the feedback characteristics of each instance. The characteristics of the feedback were analysed to identify possible important characteristics that caused feedback to have a strong impact.

Day Summaries. Each day summary was extracted from the online database and formatted into a readable useable format. Each day summary was analysed individually and notes were made on each summary. The impact at the end of each day as reported by the participants was investigated, and the aspects impacted by received feedback identified. The importance of the analysis of the day summaries was not to identify a count of the number of days with similar impacts, but to identify the delayed impact that received feedback was reported to have on the participants.

After analysing the two different forms of data collected during the diary study it was important to combine them and analyse them as a complete diary. As the participants completed 4-5 days of the diary, the analysis focused on an individual day-path analysis, and a complete week-path analysis. These approaches are discussed in the following sections

Day Path Analysis. Each day was analysed per participant by combining the day summary and all instances of feedback received during each day (example: Figure 5.8). The arrows indicate the researchers' interpretation of the participant's reported instances of feedback and their day summary. In the example (Figure 5.8), the participant reported (FB 65) a "positive" impact, and feeling "good" and "happy" after receiving an email from the CEO, which is represented by a positive arrow. In FB 63, a longer arrow is used to represent the researchers interpretation of a more influence instance of feedback, where the participants reported a "positive" impact, and said that he felt "happy", "energised", "excited", and "inspired" after receiving an email reply that was "extremely happy". The participant's response choices and additional free-text data suggest that feedback instance FB63 has more effect on the participant than feedback instance FB 65, and the figure represents that.

It was possible to view the feedback received by the participant during a day, and the reported end of day impact of the received feedback as identified during the day summary. Graphs were generated for each day to display the feedback received during that day, and the end of day summary as reported by the participant. This provided useful contextual information that aided the understanding of each participant's end of day summary in relation to the feedback they had received.

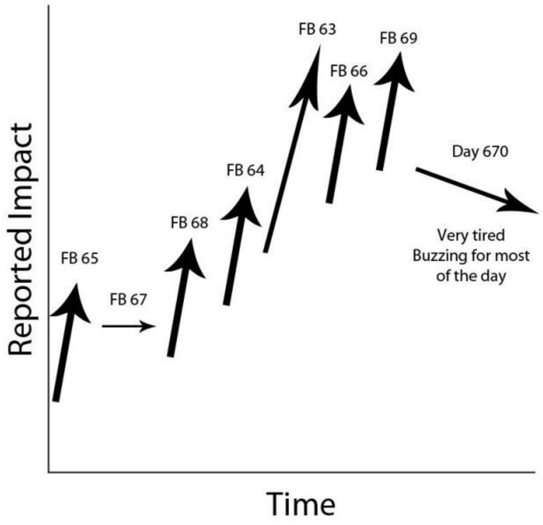


FIGURE 5.8 – DIARY DAY PATH ANALYSIS EXAMPLE

Week Path Analysis. Following from the day path analysis, it was possible to analyse the entire week for each participant (example: Figure 5.9). The styles of the analysis figures were adapted to be more representative of the data. The week path analysis was done by combining all of the day paths for each participant to form a week, allowing the investigation of feedback having an effect on following days after it had been received. Graphs were generated showing the feedback received by each participant, including instances of feedback and the day summary for each day, combined to represent one week for the participant. Similarly to the steps taken during the day path analysis and discussed in the example day path analysis, the participant's responses were analysed and are represented by arrows that are relative to the effect of the instance of received feedback or day summary.

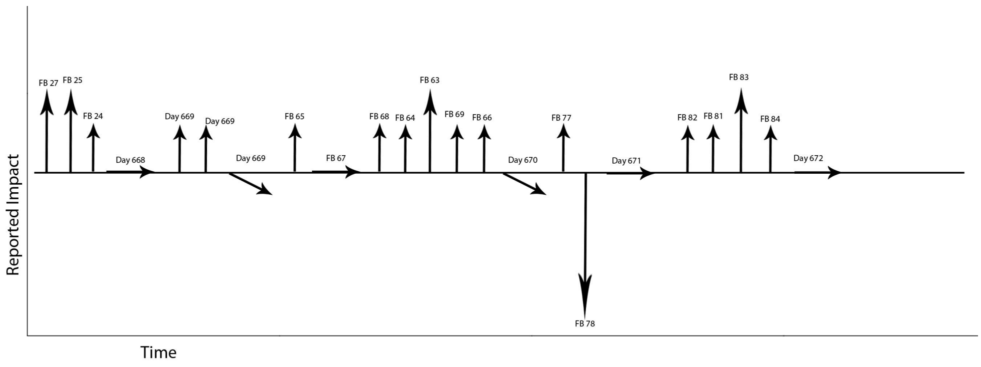


FIGURE 5.9 – DIARY WEEK PATH ANALYSIS EXAMPLE

The day path and week path analysis figures aiding understanding of how feedback could have a longer effect on software engineers, and helped illustrate the instances of feedback received by the participants during one day and during one week. All of the day path analysis figures and all of the week path analysis figures are in appendix B section 6.

5.4.3 DIRECT OBSERVATION

While the diary study was in progress, direct observation of the participants was also conducted. Focusing mainly on the 15 diary study participants, the direct observation served as a useful method to understand more about the environment in which the software engineers worked, as discussed in section 5.2, and to provide context and understanding of the participants' responses during the interview and diary studies.

DATA PREPARATION

The audio recordings from the direct observations were imported into NVIVO, and transcribed where possible. Due to background noise and the distance between the audio recording device and intended conversations, the recorded audio proved difficult to transcribe accurately.

DATA ANALYSIS

Audio recordings and written notes were collected. The written notes were used to support observations of the environment. While it had initially been planned to use observation notes to provide examples of feedback, the data collected was not suitable for this. Both the audio recordings and the written notes were not analysed further.

5.4.4 PERSONALITY INVENTORY

17 participants completed the personality inventory, including 11 of the 15 diary study participants.

DATA PREPARATION

Similar to the diary study, the personality inventory was completed online, and the data stored online. Pages were created that presented the results generated for each participant in a table, and facilitated sorting and analysis. The data was also exported and stored in a secure password-protected EXCEL workbook to aid analysis.

DATA ANALYSIS

The personality inventories for each participant were extracted and compiled with all of the participants. The compiled results were compared to other studies of software engineers to establish how representative the participants were. The personality results of each participant were used to investigate any emerging patterns in responses to questions by software engineers with similar personality results, but provided inconclusive results. The main use of the personality inventory was to establish to what extent the participants were representative of software engineers.

5.4.5 PERSONAL PROFILES

For participants who participated in multiple studies, profiles were produced to collate the data collected across all of the different research methods (example: Figure 5.30). The profiles combined extracted, themed data with a narrative of the participant that included direct quotations, providing a rich summary of all of the data collected for each participant. The profiles included data collected from the following sections, as appropriate for each participant:

- Participant demographics [I]

This section includes data collected specifying the demographics of the participant. Their years of experience, their level of education, and their experience in development environments are all demographics included in this section.

- Personality inventory results [P]

This section includes the results of the personality inventory for each participant. Their scores for each of the five factors are included, as well as a group letter used to identify participants with similar personality inventory scores.

- Work desirables (Motivators) [I]

This section includes themes that emerged from questions presented to the participant during the interview that investigated what they enjoyed and were motivated by at work. Example themes include the company culture, helping people, and building software.

- Work undesirables (De-motivators) [I]

This section includes themes that emerged from questions presented to the participant during the interview that investigated what they did not enjoy and were demotivated by at work. Example themes include unfamiliar code, obstacles, and poor management.

- Feedback in software engineering [I] [D]

This section focuses on data provided by the participant in the interview and during the diary study, and includes data discussing types of feedback typically experienced by the participant, and any thoughts they had on receiving feedback. Examples include colleague feedback, management feedback, and feedback perceived through a lack of complaints.

- Feedback impact in software engineering [I] [D]

This section identifies the impact feedback was reported to have by the participant during the interview and diary study. Examples include negative feedback impacting behaviour, and positive feedback impacting job satisfaction.

- The impact of feedback from software code compilers [I] [D]

This section focuses on the impact that feedback received from a code compiler was reported to have. Examples include that compiled code is self-affirming, and that code that doesn't compile can be frustrating and stressful.

Key: [I] Interview, [P] Personality Inventory, [D] Diary Study

Included in each personality profile is a narrative expanding on the data presented in the profile, clarifying the identified themes, and providing direct quotations from the participant showing examples of each identified theme. The combination of each profile section and the profile narrative provides a complete summary of each individual participant including data collected from all applicable research methods.

Care was taken to only include relevant quotations and to avoid enhancing or diminishing the importance of any of the factors discussed by the participants.

5.5 FINDINGS

This section presents the findings from all of the different research methods completed with the Red Gate participants as part of this study. Overall, the study found that feedback was reported to have an effect on software engineers, and that specific characteristics (the source and the medium) could change the impact of received feedback. The following section presents the findings from the analysis of the responses to each of the motivation and de-motivation factor questions from the interview in turn.

5.5.1 (DE)MOTIVATORS IN SOFTWARE ENGINEERING

The results presented below show the number of individual participants who discussed each theme in response to each question. Some participants discussed the same theme multiple times per question, but the data presented focuses on the total number of participants discussing each theme at least once, and is not a count of the total number of times a theme was mentioned. This approach illustrates the coverage of any identified theme across all the participants.

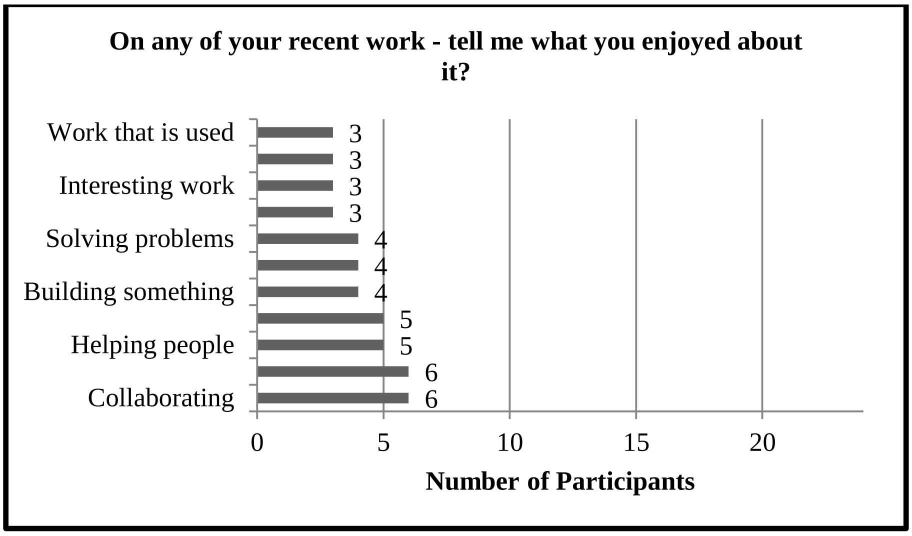


FIGURE 5.10 – THEMES EMERGING FROM RESPONSES TO QUESTION 1

Question 1 (Figure 5.10) on any of your recent work - tell me what you enjoyed about it: *Collaborating* and *improving software* were the two most frequently reported themes, each being discussed by 6 of the 24 participants. *Collaborating* with others, e.g.; “it’s quite enjoyable working with other people and see what they come up with” and “it’s been good fun, working with other people sort of doing that pair programming type stuff”. 6 participants spoke about *improving software*, improving the quality of already written software, “increasing the total amount of good code involved” and “increased the quality of the code and generally made the world happier”.

Helping people and *learning* was reported by 5 participants each. 'Helping people' included providing solutions to individuals problems "the main thing that I find enjoyable in general is just managing to solve problems for people" and "most rewarding things that I've found from software have been seeing people use what I've written and finding it useful solving problems". 5 participants reported *learning* about other systems or languages "I like learning, so when I'm finding out how new systems work, be that internal systems or third party libraries or new api's or whatever, that's always interesting" and "so working on [product] gives me a chance to sort of learn about it at that level which motivates me".

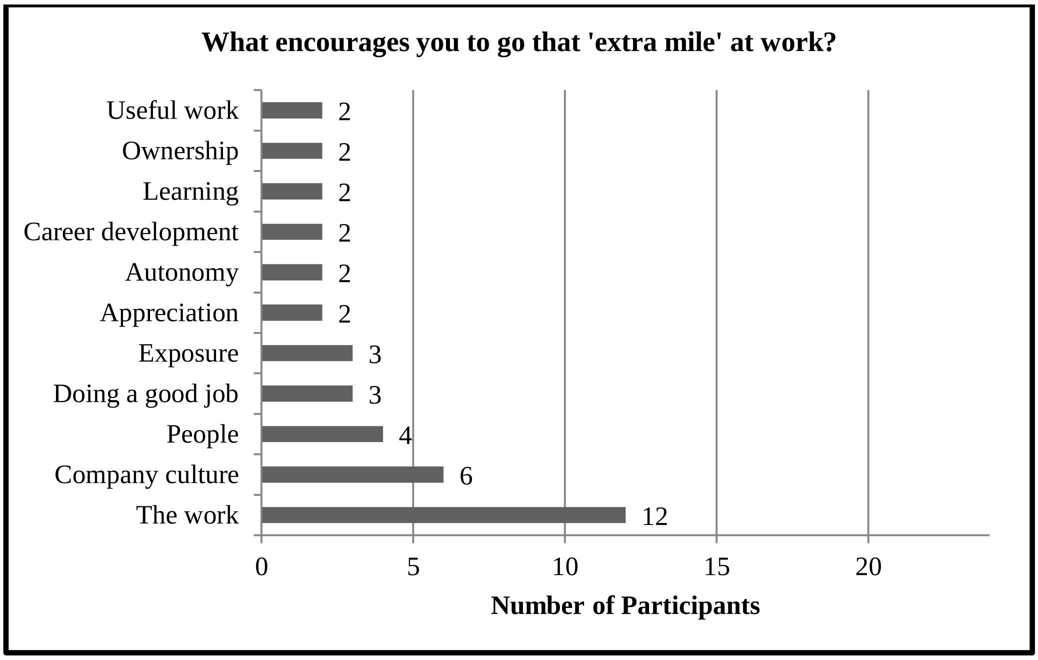


FIGURE 5.11 – THEMES EMERGING FROM RESPONSES TO QUESTION 2

Question 2 (Figure 5.11) what encourages you to go that 'extra mile' at work: *The work* was discussed by 12 of the 24 participants, e.g.: "I think it's just the combination of enjoying writing the code knowing that what you're producing is going to be cool" and "it's that thought that we're going to produce this product to a good standard and we're going to get it out when we said we're going to." *Company culture* was reported by 6 participants, and was illustrated by one participant who said "I'm still reasonably inclined if not more so to actually put in the effort here where it's needed…" "…because of the way that Red Gate generally operates in that there is a lot of flexibility, I know that for example if it's slightly more quiet time of the project, or even if it isn't, if I need to go out in the afternoon for a house viewing or something like that then there's not a problem."

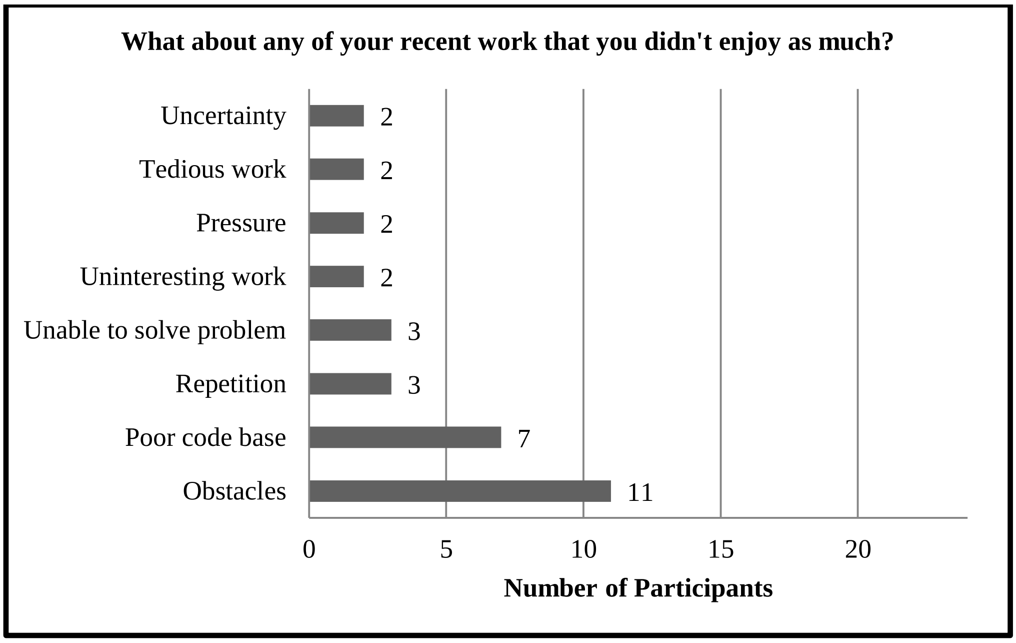


FIGURE 5.12 – THEMES EMERGING FROM RESPONSES TO QUESTION 3

Question 3 (Figure 5.12) what about any of your recent work that you didn’t enjoy as much: *Obstacles* was an emergent theme discussed by 11 of the 24 participants. One participant encapsulated *obstacles* with “your job as a software engineer is to write code and to write good code and things that make it hard are where other things get in the way”, and another participant said “anywhere where again there are outside obstacles or be it systems that are not directly under your control which are operating badly or worked and you didn’t touch anything and then they stopped working.” *Obstacles* emerged as a theme from participants responding with different obstacles that got in the way of their work.

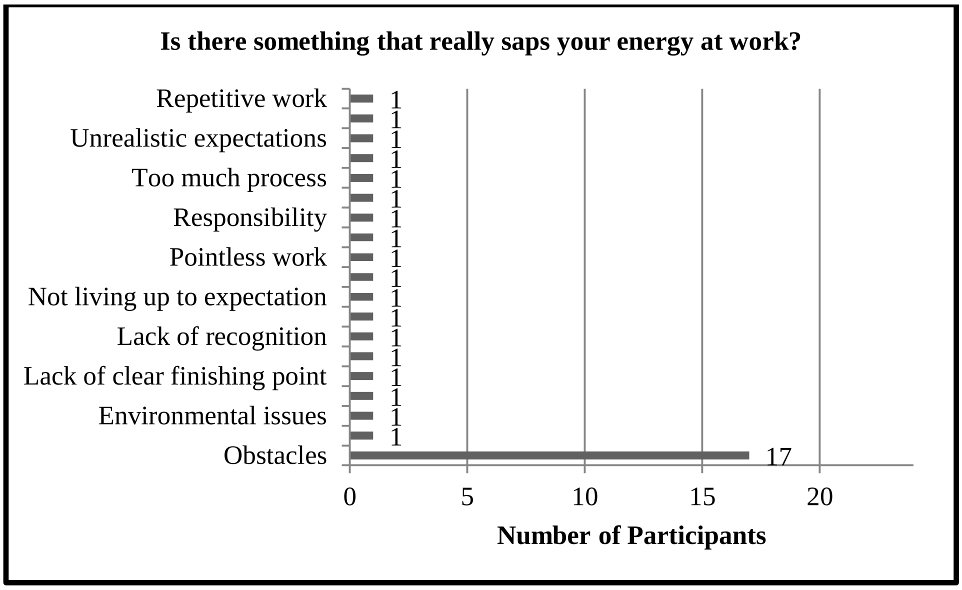


FIGURE 5.13 – THEMES EMERGING FROM RESPONSES TO QUESTION 4

Question 4 (Figure 5.13) is there something that really saps your energy at work: *Obstacles* was a frequent theme in response to question 4, discussed by 17 of the 24 participants. *Obstacles* included a range of factors including a poor code base "when you've got a large complicated poorly written code base and you have to add some small feature to it and you think well this shouldn't take very much time and it's just taking forever", lack of project direction "not really knowing where you should be going and no one setting it is always a challenge" and complicated contingencies "like you fix something, and something else over here breaks so you fix that, and something else over here breaks so you fix that and it just keeps on going round and it's horrendous."

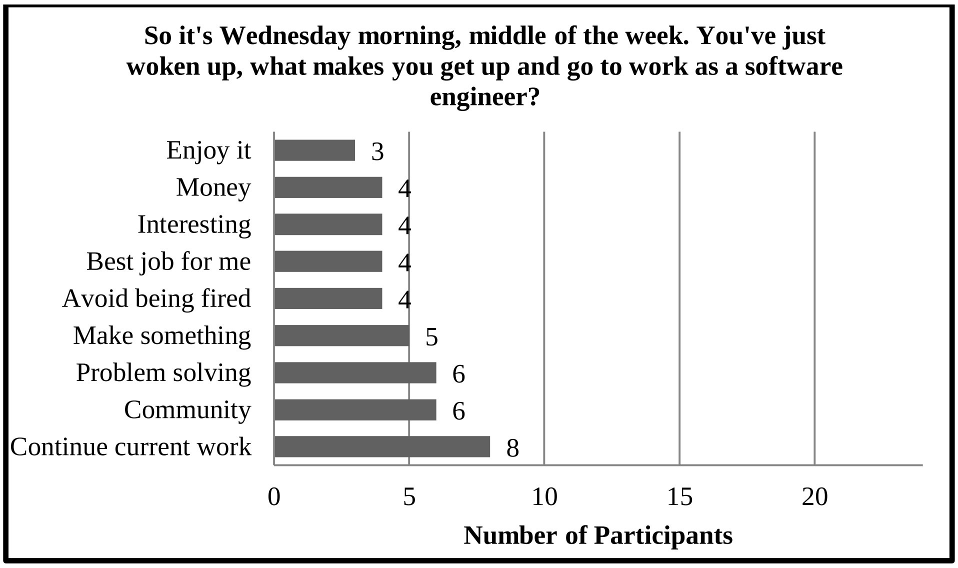


FIGURE 5.14 – THEMES EMERGING FROM RESPONSES TO QUESTION 5

Question 5 (Figure 5.14) so it's Wednesday morning, middle of the week. You've just woken up, what makes you get up and go to work as a software engineer: The participants provided a range of reasons that got them up in the morning. *Avoiding being fired* and *enjoy it* are very contrasting responses. 8 participants said they went work to continue "it's because I've left something in an incomplete point, I quite want to finish it" and "Typically because on Tuesday you left something in that isn't finished yet." 6 participants spoke about the *community* and the people at Red Gate: "I like coming in here, it's good to work with these guys, its good fun, there's a lot of banter" and "I know that I like the people, the company looks after me". 5 participants discussed *making something*: "I get to create something that's useful and it's used by lots of people to help get their jobs done" and "when I go to work I've made something by the end of the day."

5.5.2 THE OCCURRENCE AND IDENTIFICATION OF FEEDBACK IN A SOFTWARE ENGINEERING ENVIRONMENT

During the interviews, it was established that feedback existed in the software development environment and emerged from the software engineers' language prior to the term being introduced by the researcher. Software engineers were asked how they knew things were going right or wrong with their work and from their responses feedback was identified as one of the emergent themes (Figure 5.15). The themes presented here were emergent from the data provided by the participants, and each participant could provide data that falls into multiple themes.

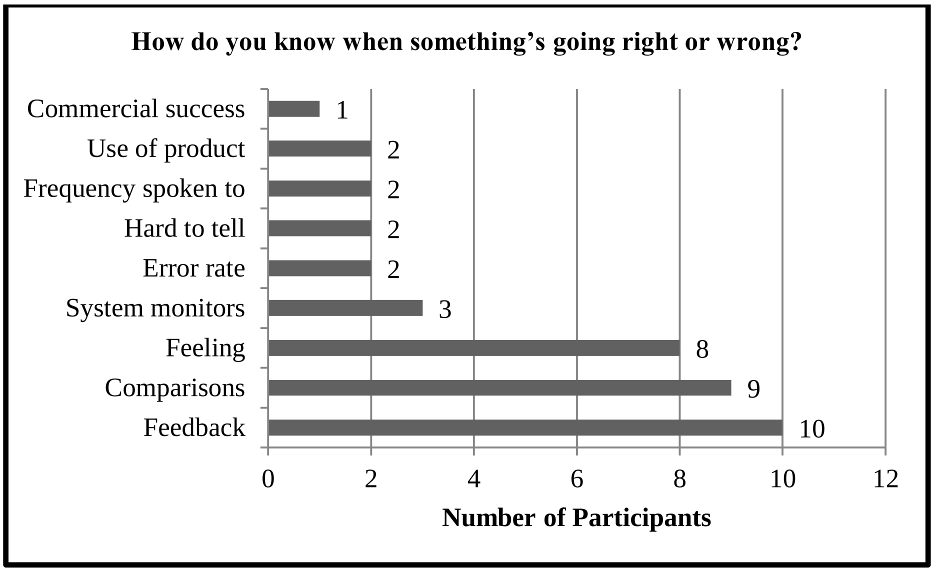


FIGURE 5.15 – EMERGENT THEMES IDENTIFYING HOW A PARTICIPANT ASSESSED PROGRESS

The investigation continued by identifying how software engineers were informed when a colleague (Figure 5.16) or manager (Figure 5.17) thought they had done a good job. For both colleagues and managers, the participants reported being told directly in person as the most common way they would receive feedback. The difference between a colleague and a manager telling them that they had done a good job was that often the manager chose to do it in a meeting setting.

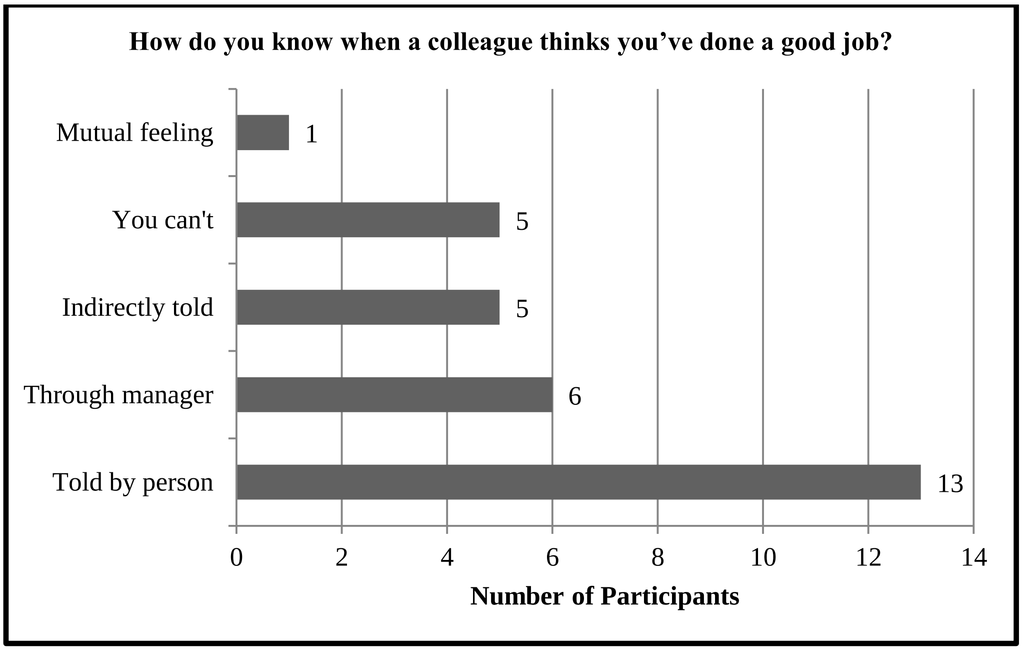


FIGURE 5.16 – EMERGENT THEMES DISCUSSING HOW A SOFTWARE ENGINEER KNOWS A COLLEAGUE THOUGHT THEY'D DONE A GOOD JOB

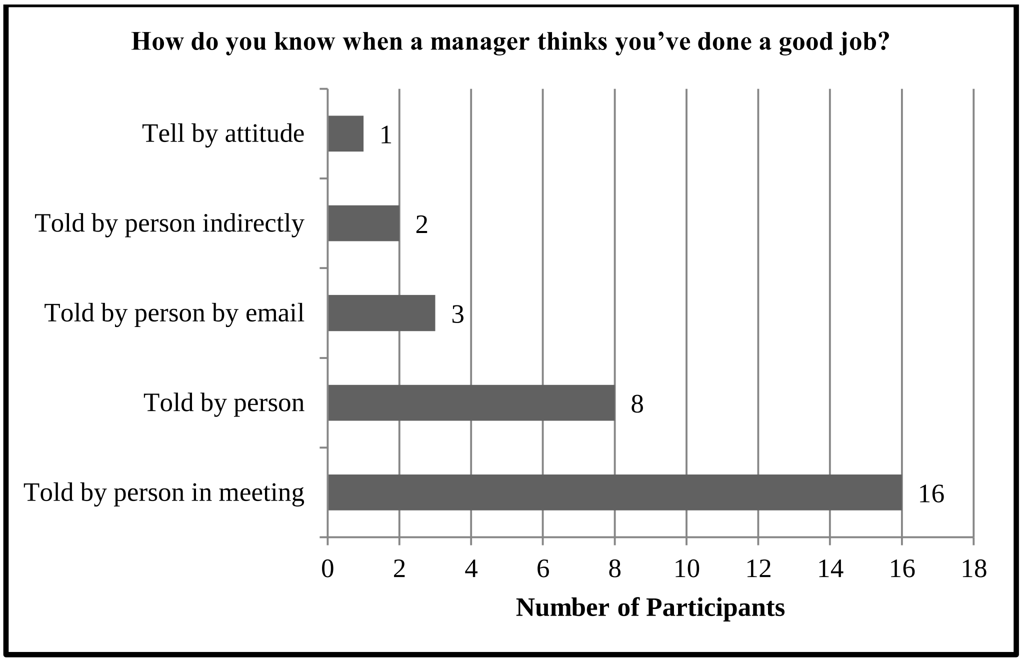


FIGURE 5.17 – EMERGENT THEMES DISCUSSING HOW A SOFTWARE ENGINEER KNOWS A MANAGER THINKS THEY HAVE DONE A GOOD JOB

The participants were asked how they would know if a colleague (Figure 5.18) or manager (Figure 5.19) thought they had done a bad job. The participants indicated that similar to the two previous questions, they received the feedback directly from a colleague or their manager.

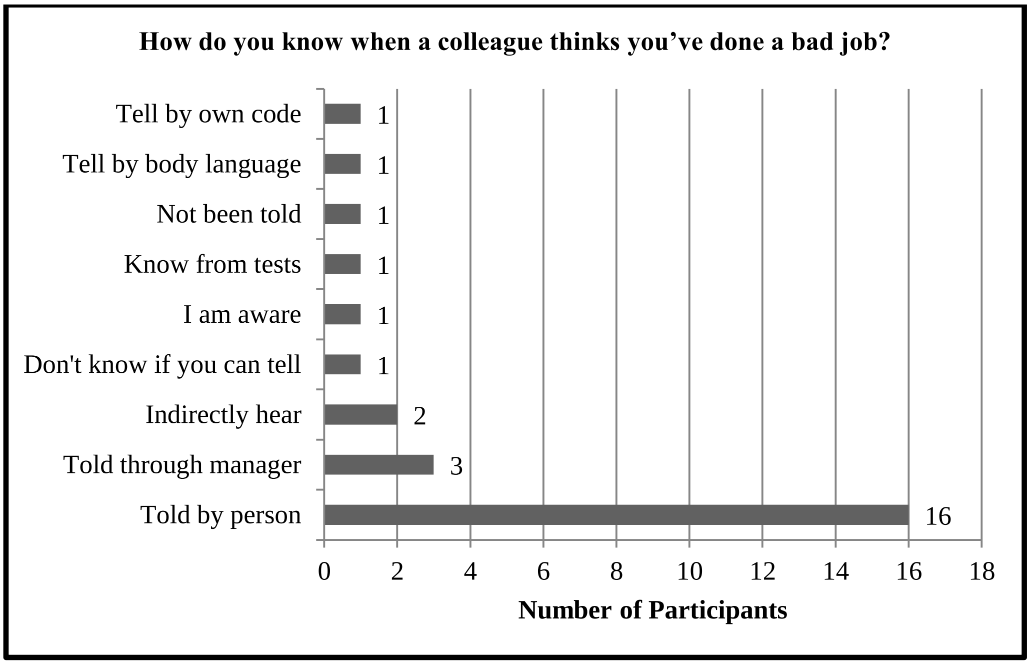


FIGURE 5.18 – EMERGENT THEMES DISCUSSING HOW A SOFTWARE ENGINEER KNOWS A COLLEAGUE THINKS THEY HAVE DONE A BAD JOB

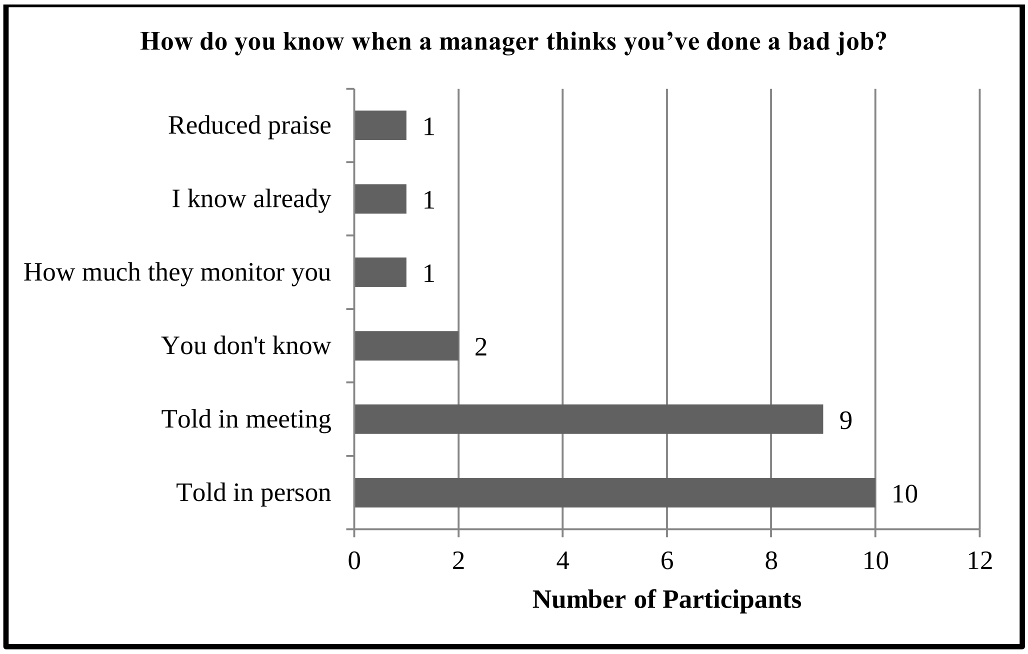


FIGURE 5.19 – EMERGENT THEMES DISCUSSING HOW A SOFTWARE ENGINEER KNOWS A MANAGER THINKS THEY HAVE DONE A BAD JOB

After completing of the interviews the study progressed onto the web-based diary study outlined in the research design (chapter 4). From the interviews it was possible to identify that feedback was being received by software engineers that contained a range of feedback characteristics with different values comparable to those identified and discussed during the Feedback Scoping Study (chapter 3). To re-cap, the characteristics included in the diary study were:

- Type – Positive or negative.
- Source – Who provided the feedback.
- Subject – What the feedback was about.
- Medium – How the feedback was received.
- Setting – The environment in which the feedback was received.

Table 5.4 presents the most frequently-reported combination of values for five feedback characteristics (type, source, subject, medium, and setting). The most frequently reported instance of feedback was positive feedback about the software engineer's performance received from a colleague by face-to-face communication in a casual setting.

There were 76 instances of feedback submitted by 15 software engineers during one week. From the 76 instances of feedback, 47 unique combinations of the values of the feedback characteristics were identified. Across all the participants, 19 forms of feedback as identified by the values of the feedback characteristics were reported multiple times by the software engineers (Table 5.4). Some of the feedback characteristics allowed the selection of multiple values, resulting in some of the 76 instances of feedback represented multiple groupings of the values of feedback characteristics.

TABLE 5.4 – REOCCURRING INSTANCES OF FEEDBACK REPORTED BY SOFTWARE ENGINEERS DURING A DIARY STUDY

Type	Source	Subject	Medium	Setting	Instances Reported
positive	colleague	performance	face-to-face	casual	7
positive	colleague	performance	email	casual	6
positive	colleague	performance	face-to-face	meeting	5
positive	colleague	attitude	face-to-face	meeting	5
positive	manager	performance	face-to-face	meeting	4
positive	manager	attitude	face-to-face	meeting	4
negative	colleague	performance	face-to-face	casual	3
positive	colleague	behaviour	face-to-face	meeting	3
positive	colleague	behaviour	face-to-face	casual	3
positive	colleague	attitude	face-to-face	casual	3
neutral	colleague	performance	face-to-face	casual	3
positive	manager	performance	email	casual	3
negative	colleague	attitude	face-to-face	meeting	2
positive	colleague	attitude	email	casual	2
positive	customer	performance	email	casual	2
negative	colleague	behaviour	face-to-face	casual	2

Type	Source	Subject	Medium	Setting	Instances Reported
positive	colleague	behaviour	email	casual	2
positive	manager	performance	face-to-face	casual	2
negative	colleague	performance	email	casual	2

Table 5.5 presents the number of times that a specific value or combination of specific values were reported for five feedback characteristics (type, source, subject, medium, and setting). In 58 instances of feedback, ‘colleague’ was reported as being the source of the feedback, and in 51 instances of feedback, ‘casual’ was reported as being the setting in which the feedback was received. The most frequently reported combination of two feedback characteristics was ‘colleague’ and ‘casual’, which were reported together in 44 instances of feedback from the 76 total instances of feedback reported during the diary study.

Some forms of feedback, as identified by the combination of the values of the feedback characteristics, were reported more frequently than others. As shown in Table 5.5, 58/76 instances of feedback were reportedly received from a colleague, and 51/76 instances of feedback were received in a casual setting. 48/76 instances of feedback were reported as being positive feedback, and 17/76 were reported as being negative instances of feedback. Table 5.5 identifies both the most frequently reported individual value of feedback characteristics, and the most frequently reported groupings of values of feedback characteristics where there were at least 10 instances of the same feedback characteristic values.

TABLE 5.5 – COMMONLY REPORTED FEEDBACK CHARACTERISTICS VALUES REPORTED BY SOFTWARE ENGINEERS DURING A DIARY STUDY

Type	Source	Subject	Medium	Setting	Instances Reported
	colleague				58
				casual	51
positive					48
			face-to-face		47
positive					47
	colleague			casual	44
	colleague		face-to-face		40
		performance			39
positive	colleague				35
			face-to-face	casual	31
positive				casual	30
	colleague		face-to-face	casual	29
	colleague	performance			29
positive			face-to-face		29
		performance		casual	28
positive		performance			27
positive	colleague			casual	25

Type	Source	Subject	Medium	Setting	Instances Reported
			email		24
positive	colleague		face to face		23
		performance	face-to-face		22
	colleague	performance		casual	22
			email	casual	20
	colleague	performance	face-to-face		19
positive		performance		casual	18
positive	colleague	performance			18
		attitude			17
	colleague		email		17
positive			email		17
negative					17
				meeting	16
	colleague		email	casual	16
positive			face-to-face	casual	16
positive		performance	face-to-face		15
positive	colleague		face-to-face	casual	15
negative					14
positive			email	casual	14
	colleague	attitude			14
	manager				14
		performance	email		14
			face-to-face	meeting	14
		performance	face-to-face	casual	14
positive				meeting	14
positive		attitude			14
positive	manager				13
		performance	email	casual	13
positive	colleague	performance		casual	13
positive			face-to-face	meeting	13
neutral					13
	colleague	performance	face-to-face	casual	13
		attitude	face-to-face		12
		behaviour			12
positive	colleague	performance	face-to-face		12
negative	colleague				12
neutral	colleague				12
positive	colleague		email		11
positive		performance	email		11
	colleague	behaviour			11
positive	colleague	attitude			11
positive		attitude	face-to-face		10
	colleague	attitude	face-to-face		10
positive		performance	email	casual	10
positive	colleague		email	casual	10
	colleague			meeting	10
	manager		face-to-face		10
neutral	colleague		face-to-face		10
neutral			face-to-face		10
negative				casual	10

These feedback characteristic values and their representativeness in the collected data identify the values of feedback characteristics most commonly reported in a software engineering environment. This suggests that the most commonly reported feedback characteristic value may also be considered by the participants to have the most impact as they decided to report them in their online diary. However, it is possible that the most commonly reported feedback characteristic is just the most prominent characteristic for the software engineers in this study. In the next section the findings of an analysis of the feedback instances reported by the participants is presented.

5.5.3 THE IMPACT OF FEEDBACK RECEIVED IN SOFTWARE ENGINEERING

During the interviews with software engineers, after they identified feedback as a factor present in their environment, the interviewer asked follow-up questions to elicit examples of feedback that the participant had received, and the impact it had on them. This approach meant that participants did not necessarily talk about the same form of feedbacks as each other. The majority of discussed feedback was presented by the participants as either positive or negative, and from either a colleague or a manager. Due to this, the remaining analysis in this section focused on investigating four identified different forms of feedback typically discussed by the participants:

1. Positive colleague feedback.
2. Positive manager feedback.
3. Negative colleague feedback.
4. Negative manager feedback.

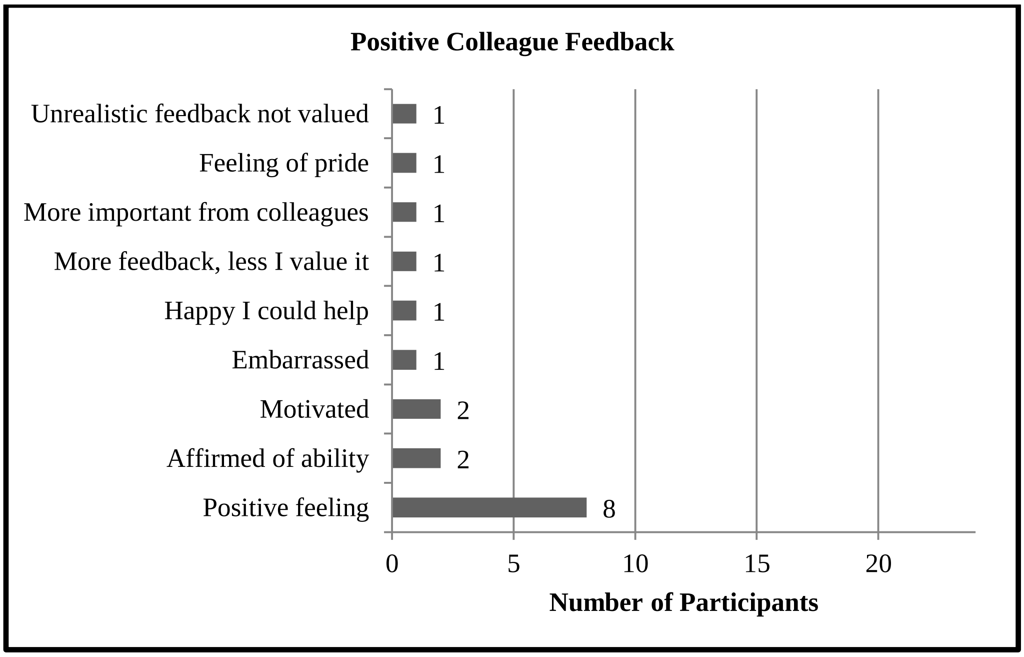


FIGURE 5.20 – THE THEMES OF IMPACT FROM POSITIVE COLLEAGUE FEEDBACK

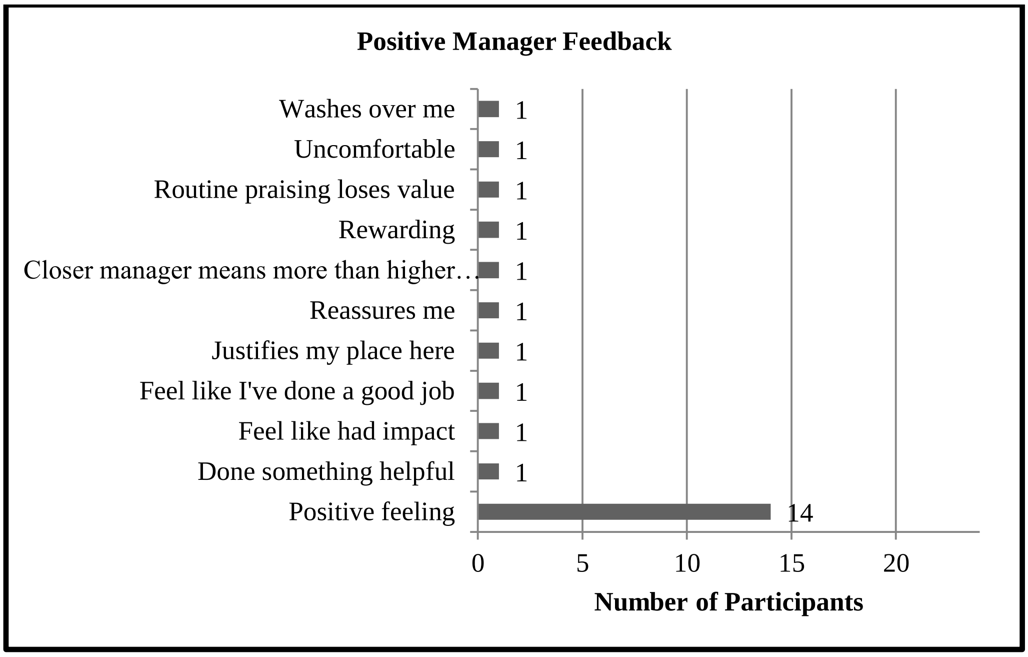


FIGURE 5.21 – THE THEMES OF IMPACT FROM POSITIVE MANAGER FEEDBACK

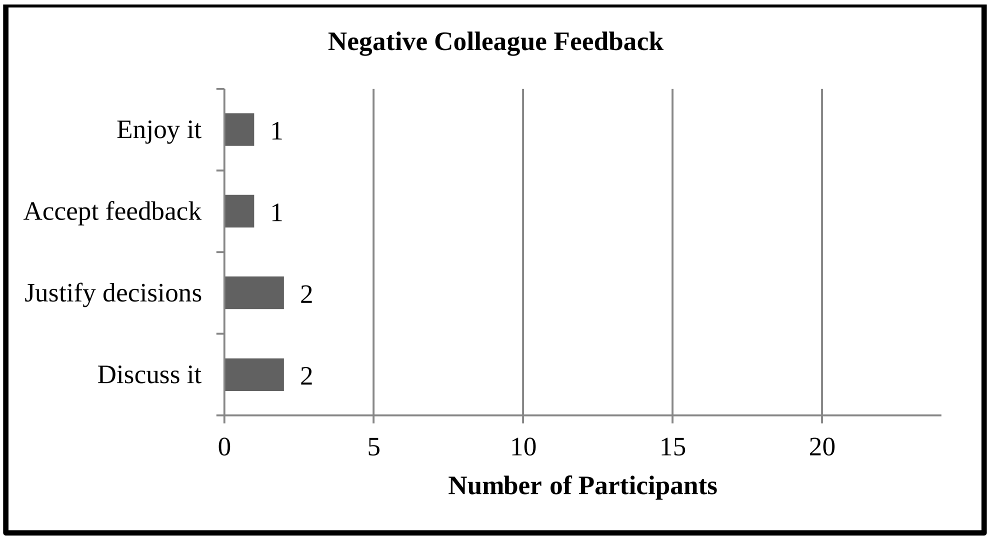


FIGURE 5.22 – THE THEMES OF IMPACT FROM NEGATIVE COLLEAGUE FEEDBACK

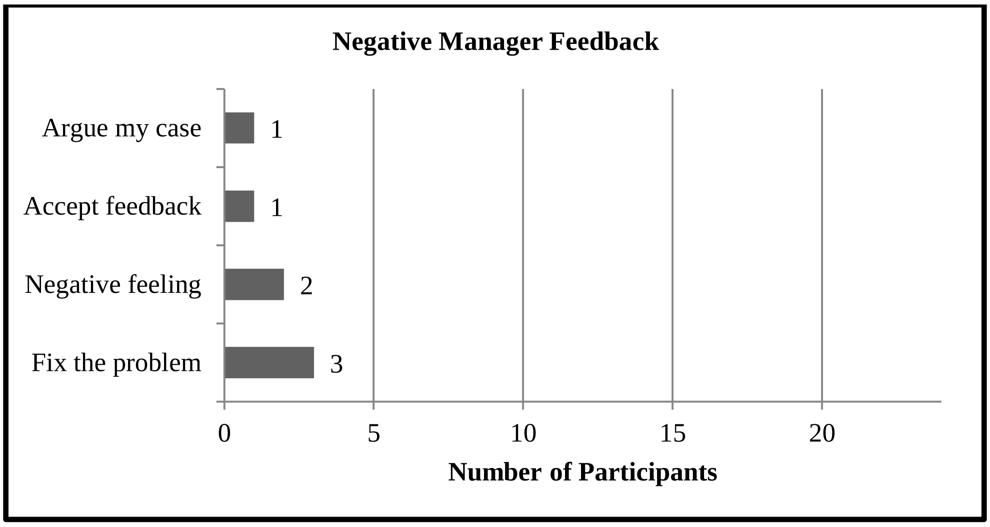


FIGURE 5.23 – THE THEMES OF IMPACT FROM NEGATIVE MANAGER FEEDBACK

Figure 5.20, Figure 5.21, Figure 5.22, and Figure 5.23 present the impact of positive and negative feedback received from colleagues and managers. Not all participants discussed the same forms of feedback, meaning that the number of participants who provided an impact for feedback in each of the four groupings presented here varies. Participants led the discussion on the impact of feedback by providing examples of feedback they had actually received, and the impact it had on them.

5.5.4 THE IMPACT OF FEEDBACK INSTANCES REPORTED DURING THE DIARY

For every instance of feedback submitted by the participants, they were asked to indicate if the received feedback had a positive, negative, or neutral impact on them when they received it, and if the feedback caused them to feel one of 15 pre-listed feelings, or another feeling, indicated through a free-text box. Of the 76 instances of feedback submitted, 47 instances were reported as having a positive impact on the participant, 8 as having a neutral impact, 17 as having a negative impact, and 4 instances had no impact reported (Figure 5.24).

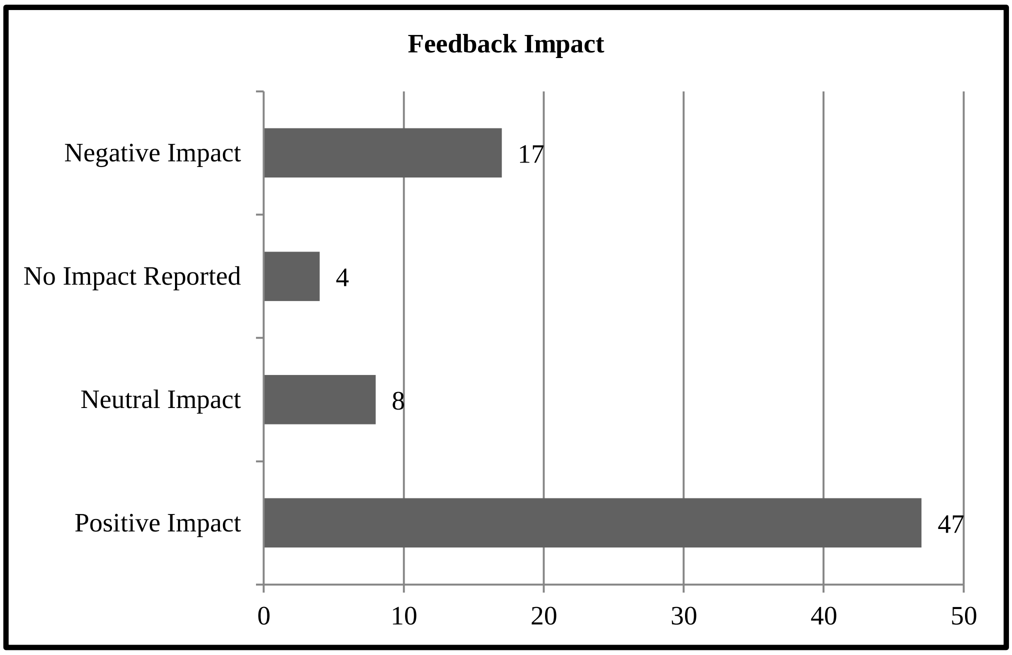


FIGURE 5.24 – FEEDBACK INSTANCE IMPACT

As can be seen in Figure 5.25, positive feedback did not always lead to a positive impact. Equally, negative feedback did not always lead have a negative impact. Some of the submitted feedback instances had no impact reported and have not been included in Figure 5.25.

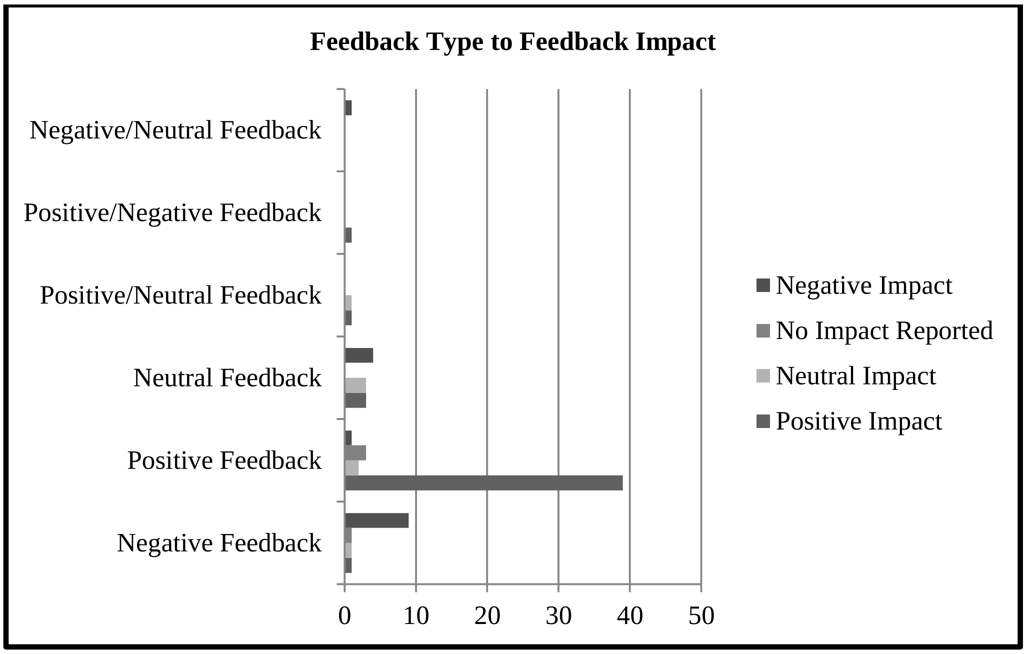


FIGURE 5.25 – FORMS OF FEEDBACK AND THEIR REPORTED IMPACT

After describing the feedback as having a positive, negative or neutral impact the participants were asked to describe how they felt after receiving the reported feedback. They were given fifteen pre-selected emotions and a free-text box should they wish to add additional emotions or comments. In 76 instances of feedback, 23 different emotions were reported, and emotions were reported as being affected by received feedback 197 times (Figure 5.26).

In general, but not always, positive feedback had a positive impact on the software engineers, and negative feedback had a negative impact. Feedback considered to be neutral had a positive, neutral, or negative impact. The findings suggest that feedback has an impact on the emotions of software engineers, and typically caused them to feel multiple emotions for each instance of received feedback, further emphasising the need to understand more about the precise impact of feedback, and the effect of the different values of feedback characteristics and how they affect the impact of feedback.

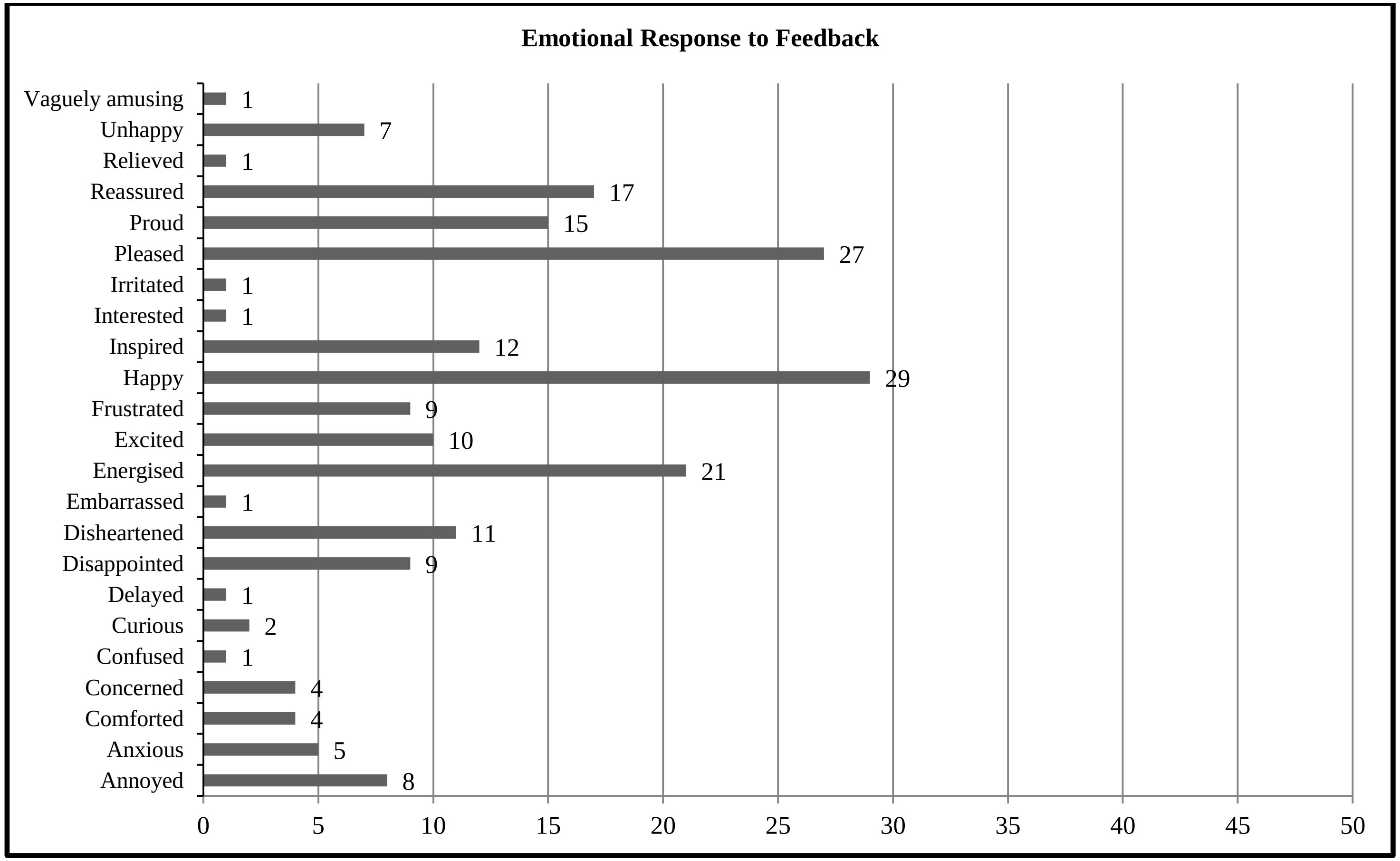


FIGURE 5.26 – REPORTED EMOTIONAL RESPONSE TO RECEIVING FEEDBACK

5.5.5 THE LASTING IMPACT OF RECEIVED FEEDBACK DISCUSSED AT THE END OF THE WORK DAY

At the end of each working day the engineers were asked to complete a day summary where they reported how they felt, if how they felt was influenced by the feedback they received that day, if the feedback they received had impacted their work, and if it had, how it had impacted their work. The engineers reported that the feedback they received did not impact on how they felt at the end of each day for 28 of the 45 completed day summaries (Figure 5.27).

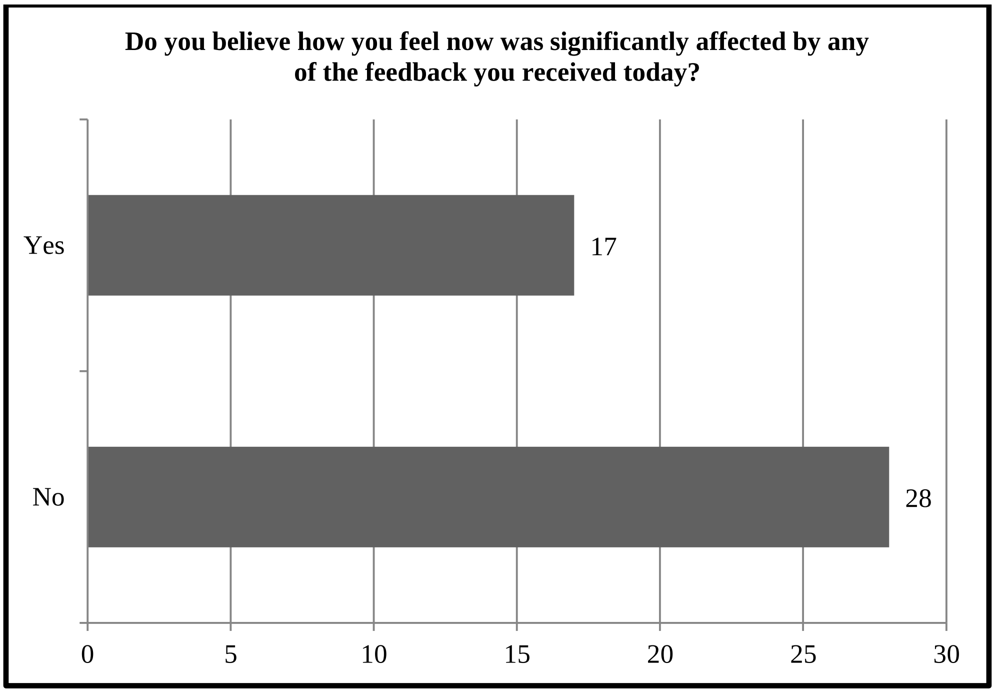


FIGURE 5.27 – DID RECEIVED FEEDBACK IMPACT A SOFTWARE ENGINEERS END OF DAY FEELING?

The engineers were asked if the feedback they had received had impacted their work (Figure 5.28), and if it had impacted their work, how had it impacted their work (Figure 5.29). These findings show that 62% (28/45) of days reported that feedback did not impact how the participant felt at the end of the day, but 49% (22/45) of the days reported that feedback had impacted the participants work that day.

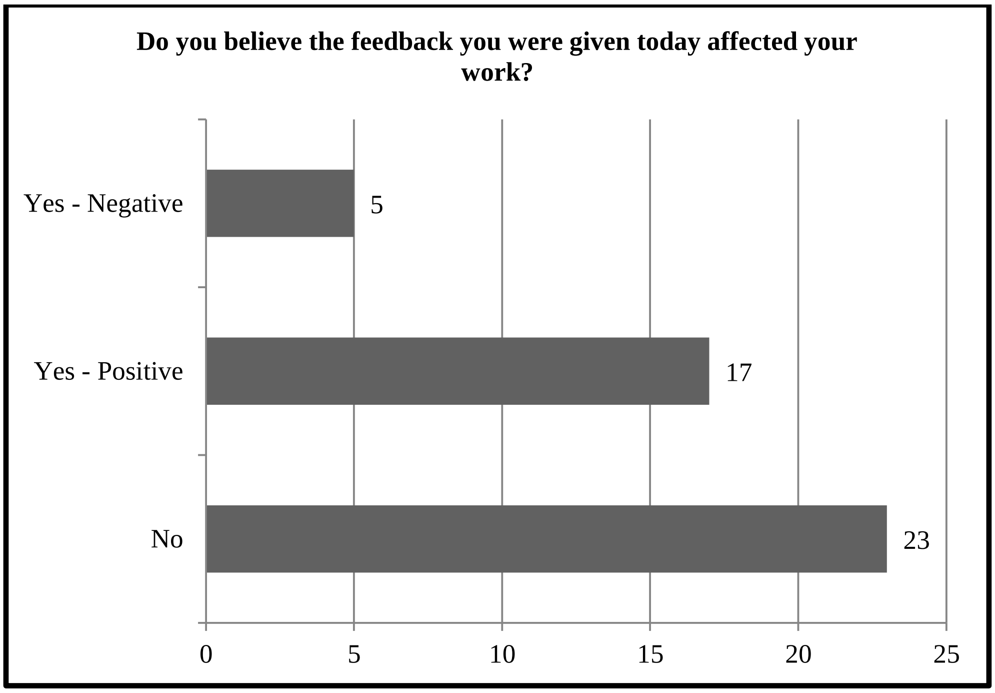


FIGURE 5.28 – DID RECEIVED FEEDBACK IMPACT A SOFTWARE ENGINEERS WORK?

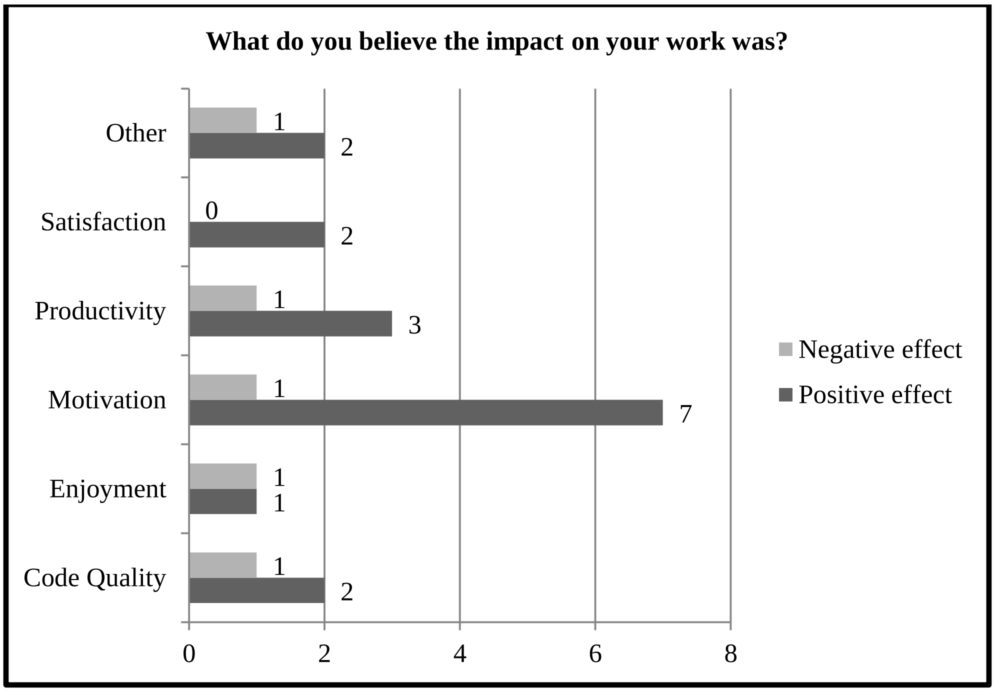


FIGURE 5.29 – HOW RECEIVED FEEDBACK AFFECTED WORK

The findings from the day summaries completed by the participants of the diary study indicate that feedback can have an impact on software engineers, extending past the initial moment of receiving feedback, and can affect how they feel at the end of the day. While most of the day summaries reported that received feedback did not have a lasting effect on how they felt, nearly 50% of the day summaries reported that the feedback had an impact, positive or negative, on their work.

5.5.6 PARTICIPANT PERSONALITIES

Of the 24 participants, 18 completed a personality inventory. The personality inventory results of the 18 participants are shown in Table 5.6. To aid analysis and understanding of the results, each participant's 'raw' score of between zero and five was calculated to be high, medium, or low. No standard approach to defining the boundaries of high, medium or low in a personality inventory measuring the five personality factors could be identified, so boundary limits had to be decided. These boundary limits affect the classification of the participants' personality scores, but after careful consideration the classification of the majority of the participants would not change without significant modifications to the set boundaries, suggesting that the set boundaries are appropriate for classifying these software engineers. The scoring for high was a score above or equal to 2.76, the scoring for low was a score below or equal to 2.24, and the scoring for medium was any score between and including 2.25 to 2.75.

TABLE 5.6 – SOFTWARE ENGINEERS PERSONALITY SCORES

User	Extraversion	Agreeableness	Conscientiousness	Openness	Neuroticism
P1	MEDIUM – 2.35	HIGH – 3.80	HIGH – 3.95	HIGH – 4.45	HIGH – 3.60
P2	LOW – 1.35	HIGH – 4.25	HIGH – 3.95	HIGH – 4.15	MEDIUM – 2.50
P3	HIGH – 3.55	HIGH – 2.80	HIGH – 3.50	HIGH – 3.30	MEDIUM – 2.50
P6	MEDIUM – 2.70	HIGH – 3.70	HIGH – 3.55	HIGH – 3.95	LOW – 2.20
P7	MEDIUM – 2.40	HIGH – 2.80	HIGH – 3.05	HIGH – 4.00	HIGH – 2.90
P9	HIGH – 2.80	HIGH – 2.80	MEDIUM – 2.55	HIGH – 3.30	HIGH – 2.85
P10	HIGH – 3.70	HIGH – 3.90	HIGH – 3.35	HIGH – 3.50	LOW – 1.90
P15	MEDIUM – 2.70	HIGH – 3.65	HIGH – 3.20	HIGH – 3.25	HIGH – 2.90
P16	HIGH – 3.05	HIGH – 3.60	HIGH – 3.95	HIGH – 3.85	LOW – 1.60
P17	HIGH – 3.20	HIGH – 3.60	HIGH – 3.75	HIGH – 3.65	HIGH – 3.05
P19	HIGH – 3.40	MEDIUM – 2.65	MEDIUM – 2.65	HIGH – 3.80	HIGH – 2.95
P20	HIGH – 4.65	HIGH – 3.80	HIGH – 4.00	HIGH – 4.80	LOW – 2.05
P21	MEDIUM – 2.50	HIGH – 3.80	HIGH – 4.05	HIGH – 3.55	LOW – 2.20
P22	HIGH – 3.85	HIGH – 3.95	HIGH – 4.35	HIGH – 3.90	LOW – 1.90
P24	HIGH – 3.15	HIGH – 3.80	HIGH – 3.60	HIGH – 3.75	HIGH – 2.90
P25	LOW – 1.15	HIGH – 3.30	HIGH – 4.15	HIGH – 3.10	HIGH – 3.20
P26	MEDIUM – 2.65	HIGH – 3.55	HIGH – 3.50	HIGH – 3.55	HIGH – 2.90
P28	HIGH – 4.05	HIGH – 4.20	HIGH – 3.75	HIGH – 3.95	HIGH – 3.20

All of the participants scored highly in openness, and the range for openness varied by a score of up to 1.35, with the lowest score being 3.10 and the highest being 4.45. All but one participant scored high for agreeableness, with the only medium score. The range of scores for participants scoring high in agreeableness was by a score of up to 1.45, from a score of 2.80 to a score of 4.25.

Participants were found to have similar scores for conscientiousness, with 2 participants reporting a medium scoring and all other 16 participants reporting a high scoring. The range of the 16 participants reported as scoring high in conscientiousness was just 1.10, with the lowest score of 3.05 and the highest score of 4.15.

Extraversion and neuroticism provided the highest variation of results across the 18 participants. Extraversion was the factor with the highest range. Two participants scored low in extraversion, with the lowest scoring just 1.15, six participants reported a medium scoring, and ten participants scored high, with the highest scoring reported being 4.65, a total range of 3.50 from the lowest scoring participant of 1.15. For neuroticism, six participants reported a low scoring, with the lowest score found to be 1.60, two participants reported a medium scoring, and ten participants reported a high scoring, with the highest score found to be 3.60, giving an overall range from the highest-to-lowest reported scorings from the 18 participants of 2.00.

Personality data was collected from the participants firstly to allow for direct participant comparisons, but also to justify that the participants in this study were representative software engineers. As identified in section 4.3, the personality inventory used was the exact same as the MyPersonality inventory administered using a similar web-based online form to over 3.3 million people.

TABLE 5.7 – PARTICIPANT REPRESENTATIVENESS

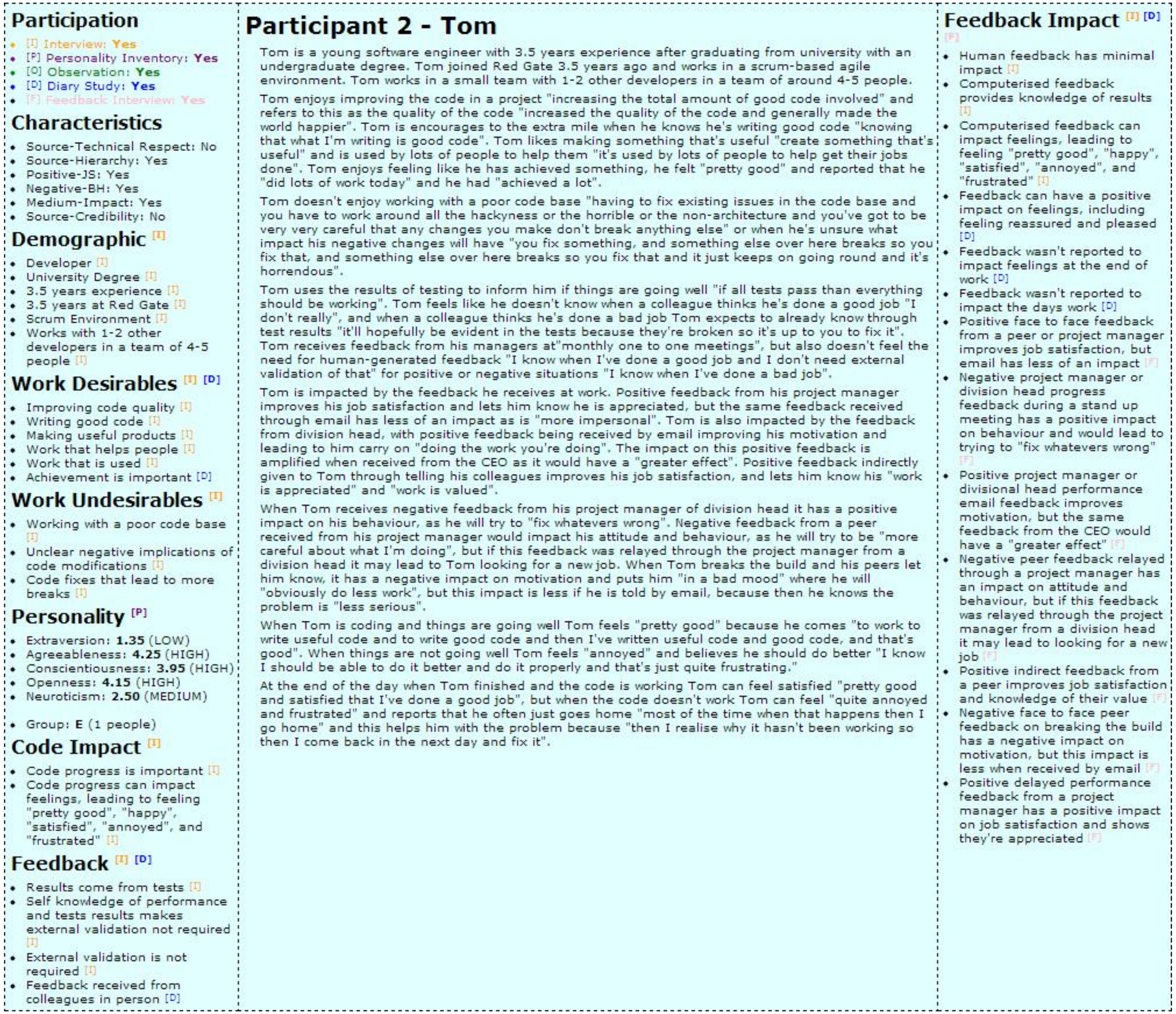
User	Extraversion	Agreeableness	Conscientiousness	Openness	Neuroticism
18 Participant Mean:	HIGH – 2.96	HIGH – 3.55	HIGH – 3.60	HIGH – 3.77	MEDIUM – 2.63
324 MyPersonality Results:	HIGH – 3.10	HIGH – 3.51	HIGH – 3.56	HIGH – 3.95	MEDIUM – 2.54

The mean averages of the 18 participants were compared to the results of the MyPersonality online survey. The subset of the 3.3million MyPersonality results was extracted using set criteria to ensure that a fair comparison was made. All respondents to the MyPersonality inventory needed to have completed the full 100-question version of the personality inventory, they had to report their work title as ‘developer’, ‘programmer’, ‘software developer’ or ‘software engineer’, and they had to complete the survey within two attempts. This produced a total of 324 suitable personality inventory results for comparison.

The comparison of the personality results of the 18 participants with the 324 suitable respondents to the MyPersonality inventory can be seen in Table 5.7. These results indicate a very small range between each of the 5 factors between each group, with the largest range of 0.22 reported in the factor openness, and the lowest range of 0.04 reported in the factor agreeableness. These results support the claim that the participants are representative of software engineers.

5.5.7 PARTICIPANT PROFILES

As described in section 5.4.5, a profile was compiled for each participant containing both themes emergent from their data, and narrative with direct quotations of what they said during data collected in any research method used during this study. An example profile for one software engineer is displayed in Figure 5.30, and all the profiles are in appendix B section 5.



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Interview: **Yes**

Characteristics

- Source-Technical Respect: No
- Source-Hierarchy: Yes
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: Yes
- Source-Credibility: No

Demographic [I]

- Developer [I]
- University Degree [I]
- 3.5 years experience [I]
- 3.5 years at Red Gate [I]
- Scrum Environment [I]
- Works with 1-2 other developers in a team of 4-5 people [I]

Work Desirables [I] [D]

- Improving code quality [I]
- Writing good code [I]
- Making useful products [I]
- Work that helps people [I]
- Work that is used [I]
- Achievement is important [D]

Work Undesirables [I]

- Working with a poor code base [I]
- Unclear negative implications of code modifications [I]
- Code fixes that lead to more breaks [I]

Personality [P]

- Extraversion: **1.35** (LOW)
- Agreeableness: **4.25** (HIGH)
- Conscientiousness: **3.95** (HIGH)
- Openness: **4.15** (HIGH)
- Neuroticism: **2.50** (MEDIUM)

- Group: **E** (1 people)

Code Impact [I]

- Code progress is important [I]
- Code progress can impact feelings, leading to feeling "pretty good", "happy", "satisfied", "annoyed", and "frustrated" [I]

Feedback [I] [D]

- Results come from tests [I]
- Self knowledge of performance and tests results makes external validation not required [I]
- External validation is not required [I]
- Feedback received from colleagues in person [D]

Participant 2 - Tom

Tom is a young software engineer with 3.5 years experience after graduating from university with an undergraduate degree. Tom joined Red Gate 3.5 years ago and works in a scrum-based agile environment. Tom works in a small team with 1-2 other developers in a team of around 4-5 people.

Tom enjoys improving the code in a project "increasing the total amount of good code involved" and refers to this as the quality of the code "increased the quality of the code and generally made the world happier". Tom is encourages to the extra mile when he knows he's writing good code "knowing that what I'm writing is good code". Tom likes making something that's useful "create something that's useful" and is used by lots of people to help them "it's used by lots of people to help get their jobs done". Tom enjoys feeling like he has achieved something, he felt "pretty good" and reported that he "did lots of work today" and he had "achieved a lot".

Tom doesn't enjoy working with a poor code base "having to fix existing issues in the code base and you have to work around all the hackyness or the horrible or the non-architecture and you've got to be very very careful that any changes you make don't break anything else" or when he's unsure what impact his negative changes will have "you fix something, and something else over here breaks so you fix that, and something else over here breaks so you fix that and it just keeps on going round and it's horrendous".

Tom uses the results of testing to inform him if things are going well "if all tests pass than everything should be working". Tom feels like he doesn't know when a colleague thinks he's done a good job "I don't really", and when a colleague thinks he's done a bad job Tom expects to already know through test results "it'll hopefully be evident in the tests because they're broken so it's up to you to fix it". Tom receives feedback from his managers at"monthly one to one meetings", but also doesn't feel the need for human-generated feedback "I know when I've done a good job and I don't need external validation of that" for positive or negative situations "I know when I've done a bad job".

Tom is impacted by the feedback he receives at work. Positive feedback from his project manager improves his job satisfaction and lets him know he is appreciated, but the same feedback received through email has less of an impact as is "more impersonal". Tom is also impacted by the feedback from division head, with positive feedback being received by email improving his motivation and leading to him carry on "doing the work you're doing". The impact on this positive feedback is amplified when received from the CEO as it would have a "greater effect". Positive feedback indirectly given to Tom through telling his colleagues improves his job satisfaction, and lets him know his "work is appreciated" and "work is valued".

When Tom receives negative feedback from his project manager of division head it has a positive impact on his behaviour, as he will try to "fix whatevers wrong". Negative feedback from a peer received from his project manager would impact his attitude and behaviour, as he will try to be "more careful about what I'm doing", but if this feedback was relayed through the project manager from a division head it may lead to Tom looking for a new job. When Tom breaks the build and his peers let him know, it has a negative impact on motivation and puts him "in a bad mood" where he will "obviously do less work", but this impact is less if he is told by email, because then he knows the problem is "less serious".

When Tom is coding and things are going well Tom feels "pretty good" because he comes "to work to write useful code and to write good code and then I've written useful code and good code, and that's good". When things are not going well Tom feels "annoyed" and believes he should do better "I know I should be able to do it better and do it properly and that's just quite frustrating."

At the end of the day when Tom finished and the code is working Tom can feel satisfied "pretty good and satisfied that I've done a good job", but when the code doesn't work Tom can feel "quite annoyed and frustrated" and reports that he often just goes home "most of the time when that happens then I go home" and this helps him with the problem because "then I realise why it hasn't been working so then I come back in the next day and fix it".

Feedback Impact [I] [D] [F]

- Human feedback has minimal impact [I]
- Computerised feedback provides knowledge of results [I]
- Computerised feedback can impact feelings, leading to feeling "pretty good", "happy", "satisfied", "annoyed", and "frustrated" [I]
- Feedback can have a positive impact on feelings, including feeling reassured and pleased [D]
- Feedback wasn't reported to impact feelings at the end of work [D]
- Feedback wasn't reported to impact the days work [D]
- Positive face to face feedback from a peer or project manager improves job satisfaction, but email has less of an impact [F]
- Negative project manager or division head progress feedback during a stand up meeting has a positive impact on behaviour and would lead to trying to "fix whatevers wrong" [F]
- Positive project manager or divisional head performance email feedback improves motivation, but the same feedback from the CEO would have a "greater effect" [F]
- Negative peer feedback relayed through a project manager has an impact on attitude and behaviour, but if this feedback was relayed through the project manager from a division head it may lead to looking for a new job [F]
- Positive indirect feedback from a peer improves job satisfaction and knowledge of their value [F]
- Negative face to face peer feedback on breaking the build has a negative impact on motivation, but this impact is less when received by email [F]
- Positive delayed performance feedback from a project manager has a positive impact on job satisfaction and shows they're appreciated [F]

FIGURE 5.30 – EXAMPLE SOFTWARE ENGINEER PROFILE

From these profiles, six participant characteristics were identified. These characteristics were used to identify each participant's preferences on six key differences identified during the data analysis. These six key differences are listed and defined below:

- Source – Technical Respect:
    - Does the participant consider and value the technical knowledge of the source when receiving feedback?
- Source – Hierarchy:
    - Does the participant consider and value the hierarchical position of the source when receiving feedback?
- Source – Credibility:
    - Does the participant consider the credibility of the source when receiving feedback?
- Positive Feedback – Job Satisfaction:
    - Does the participant report that positive feedback impacts their job satisfaction?
- Negative Feedback – Behaviour Change:
    - Does the participant report that negative feedback causes a change in their behaviour?
- Medium – Impact:
    - Does the participant report that the medium in which they receive feedback can change the impact of the feedback?

A comparison of the discussed participant characteristics enabled the identification of participants who had one or more similar characteristics to another participant. After identifying any groupings, their data were compared to see if they had any other common responses. After analysing the data, there were no consistent similarities across the participants found by comparing one or multiple participant characteristics. This may be due to an insufficient number of participants for attempting to compare several individual characteristics and multiple responses. While this did not provide a successful grouping of individual characteristics, it was a valuable step in understanding the important findings of the study, and ultimately the direction of further investigations.

5.6 THREATS TO VALIDITY

The findings of this study are subject to four main threats to validity. First, collecting data from a single organisation is an external threat to validity and limits the generalisability of the findings. While this threat to validity cannot be reduced, the threat itself provided many positive factors that were considered desirable in the design of this research: the single environment removes the consideration of cross-environment factors; the access to the environment and to multiple software engineers in one company provides extra information that would not be gathered from software engineers working in several different organisations; and because the Red Gate environment (including both management and engineering culture and practices) includes explicit attention to reflection and motivation and encourages personnel to engage frankly in activities, it might be expected that the Red Gate personnel are likely to report candidly and reflectively, and focus on what motivates them as software engineers.

Second, the sample is opportunistic; this may bias the findings as the participants may not be a representative sample of software engineers, affecting the external validity of the findings. This threat is limited, as demographic details of the participants were collected, and personality data collected from the participants supported that these software engineers were representative of the wider community.

Third, despite the fact that these software engineers were representative of the wider community, the number of participants and their environment means that these findings cannot be viewed as representative, but as indicative. While data was collected from 24 software engineers, and collecting data from one software engineering environment removed cross-environment issues, the findings may be specific to this one environment, rather than all software engineers, limiting the generalizability, and the external validity of the findings.

Fourth, the research approach taken focuses on eliciting the thoughts, perceptions, and reflections of software practitioners, as there is no means of directly observing the impact of feedback on motivation. This provides a threat to the internal validity of the findings. This threat is limited by the design of the research that includes research method triangulation, investigating feedback at different points in time, and supporting the confidence on the reported findings.

5.7 SUMMARY

Emerging from the data are a set of indications of the role that feedback plays in the motivation of software engineers, the importance and effect of the values of feedback characteristics, and common motivators and demotivators reported by software engineers. Clear groupings of participants were not found, but indicators suggesting the significant factors affecting the impact of received feedback were identified. The following sections summarise the results of this study.

5.7.1 THE WORK, PEOPLE AND OBSTACLES

Aspects of the work was the most frequent theme emerging from responses to the questions investigating the enjoyable and potentially motivating areas of software engineering. Improving software (6/24), building software (4/24), producing good software (4/24), and interesting work (3/24) were factors reported as being enjoyable. 12 of the 24 participants reported that the work encouraged them to go the extra mile. This is not a novel motivator, as *technically challenging work* was found to be the most commonly reported motivator for software engineers by Beecham *et al.* (2008).

Obstacles, things that get in the way of the work, was the most frequent theme emerging from questions investigating the less enjoyable and potentially de-motivating areas of software engineering. 11/24 participants reported it as something they didn’t enjoy about their recent work, and 17/24 participants said it sapped their energy at work. *Obstacles* encompasses a range of different factors including being disrupted during work, being held back by other team members, lack of direction during a project, and dealing with poorly written code. *Obstacles* were not identified as a factor in the literature, but some of the themes included in o*bstacles*, for example poor communication and poor management, were identified in a systematic literature review by Beecham *et al.* (2008). *Obstacles* hamper the motivational potential of *the work* by getting in the way and slowing progress, but are not necessarily de-motivators, per se.

Many different reported factors including *collaborating*, *helping people*, *company culture*, and *community* are all people factors, focusing on fellow team members, fellow company members, or customers. The importance of people in software engineering is becoming more apparent (Sharp and Hall 2009, Sach *et al.*, 2010). It is not clear how important people are, but two of the three most cited factors that software engineers reported made them enjoy their work, and two of the three most cited factors that software engineers said encouraged them to go the extra mile were ‘people’ factors.

5.7.2 FEEDBACK IN SOFTWARE ENGINEERING

Feedback was reported as the most common method used by the software engineers to know if things were going right or wrong. When the software engineers had done a good job, they would typically know by direct feedback from someone, be it a colleague or a manager. Equally, if the software engineers had done a bad job, they would typically be directly told by the person who thought they’d done a bad job. This suggests that the environment, the developers, their colleagues, and their managers are all willing to provide direct feedback to each other on a regular basis.

The software engineers received a range of feedback with different characteristic values during their work day. The most commonly cited feedback source was a colleague. From the 76 instances of reported feedback in the diary study, 63% (48) were positive, 18.6% (14) were negative, and 17.3% (13) were neutral. Of the 76 instances of feedback reported there were 19 different combinations of values of feedback characteristics reported at least twice, with the three most commonly reported forms of feedback being reported as feedback about performance from a colleague face to face (7), via email (6), and in a meeting (5). These findings suggest that the majority of feedback that the software engineers received and considered important enough to be reported was feedback about performance received from a colleague.

5.7.3 FEEDBACK IMPACT

The participants reported that positive feedback would typically have a positive impact on their feelings, and that negative feedback would lead them to want to discuss and fix the problem. In the Red Gate environment, where close-to-event feedback is encouraged, the software engineers reported that they often received positive feedback from their manager in a meeting, but negative feedback would typically be delivered face-to-face in a casual setting. Feedback from fellow software engineers would typically be given directly, not through a manager or another colleague.

Of the 76 instances of feedback recorded in the diary study, 47 instances had a positive impact. While negative feedback typically had a negative impact, and positive feedback typically had a positive impact, this was not always the case. For some engineers, repeated positive feedback would eventually lead to a neutral or negative impact, as it became repetitive, boring or lost its value.

The participants reported 23 distinct emotions being affected a total of 197 times as the direct result of received feedback. However, at the end of the day less than 40% of day summaries said that the feedback they had received affected how they felt at the end of the working day. This suggests that the participants believe that the feedback they receive has a strong impact when they receive it, but by the end of the day its impact is almost forgotten.

Just under half of the completed day summaries report that the feedback software engineers received during the day affected their work. Code quality, enjoyment, motivation, productivity and job satisfaction are all reported as being affected by feedback the software engineers have received. The specific impact on these factors by each individual instance of feedback is unclear, as this was not recorded or identified during this study.

5.8 CONCLUSION

This study investigated feedback in software engineering with software engineers in one environment, Red Gate, a software development company based in Cambridge employing over 30 software engineers and 200 employees. The study used research method triangulation to elicit the perceptions of software engineers. Using semi-structured interviews, diary studies, participant observation and personality inventories the study investigated feedback in software engineering with 24 participants.

The findings indicate that the work is motivating for software engineers, working with and helping people is enjoyable, and that things that get in the way of the work are de-motivating. Feedback happens frequently in software development environments, and often originates from colleagues providing positive feedback on the feedback recipients’ performance.

Feedback can affect how a software engineer feels at the time he/she receives the feedback, and was reported in 76 instances of feedback to affect 197 emotions, including 23 distinct emotions. Feedback can have a lasting effect on how a software engineer feels at the end of the working day. The software engineers reported that feedback could affect their work both positively and negatively, and could affect their code quality, enjoyment, motivation, productivity and job satisfaction.

These findings provide an initial indication of the impact of feedback in software engineering environments. It is clear that feedback can have a strong impact on software engineers, affecting how they feel, and affecting their work. The impact of each individual instance of feedback remains unclear, and while an impact on the participant's feelings was indicated, it was not clear how this would influence their work, or have a longer impact on their motivation.

5.8.1 QUESTIONS RAISED FROM FINDINGS

The main questions raised by this study are:

- Can the impact of an individual instance of feedback be better identified?
- What is the impact of a change in the value of a feedback characteristic?
- Are there more or less significant feedback characteristics?
- If more or less significant feedback characteristics exist, are they the same for all software engineers?

These questions will be addressed in the next study, presented in chapter 6.

5.9 SO WHAT DO WE KNOW NOW?

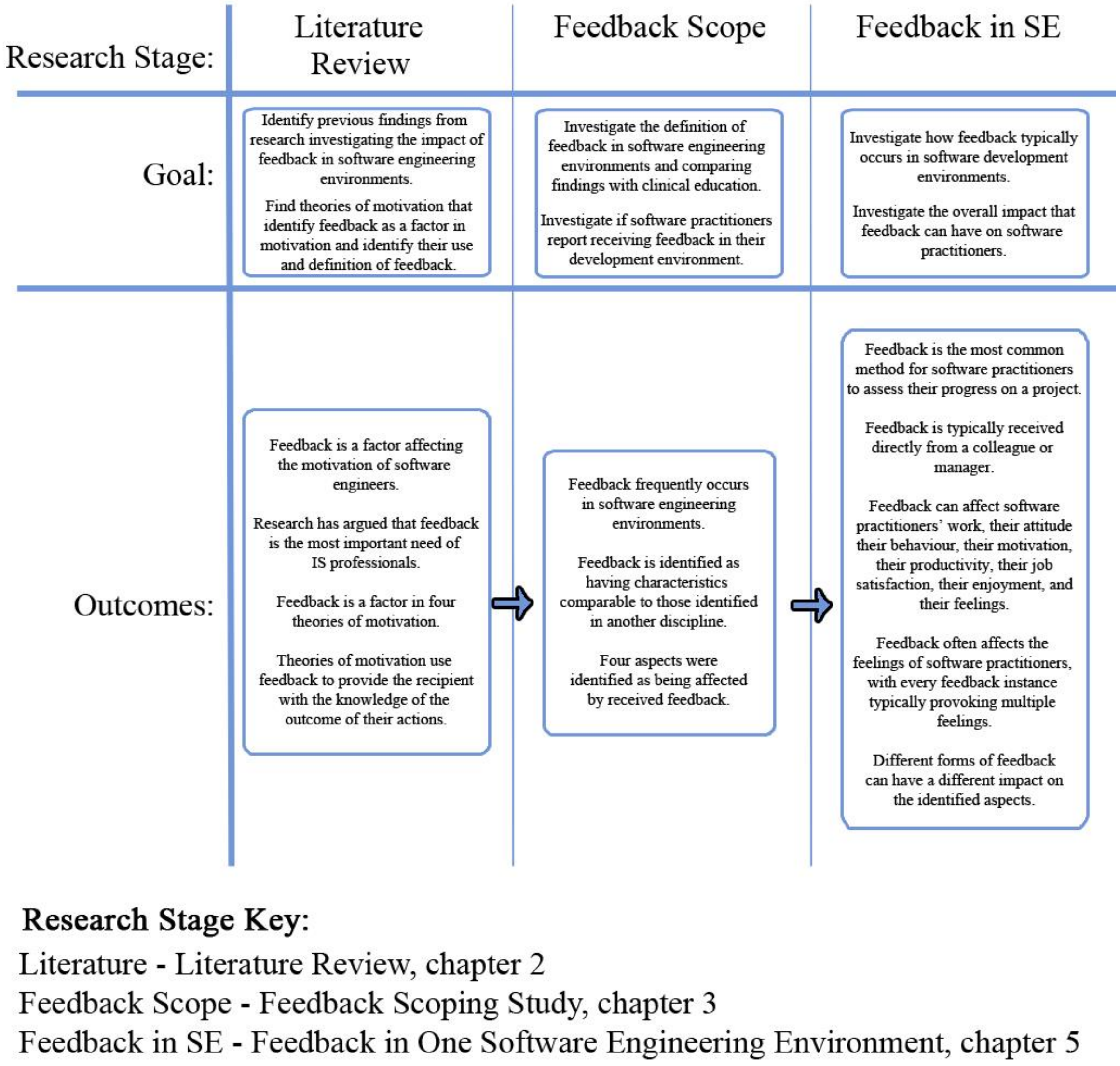
The findings from this study indicate that feedback is identified by the software engineers as their most common tool for checking the progress of their work. The feedback software engineers receive most commonly comes from fellow software engineers directly, but can also come directly from managers. Positive feedback reportedly occurs more often than negative feedback, and feedback is most commonly received in a casual environment.

Feedback can have an impact on a software engineer's work, specifically their code quality, productivity, motivation, job satisfaction, and enjoyment. Feedback can also impact the emotional state of software engineers both at the time of receiving feedback and as reported at the end of the working day. No relationship was identified between feedback instances and the impact they had on a software engineer’s work.

Building on the Feedback Scoping Study (chapter 3) findings, it is now known that feedback happens in software engineering environments and is defined with characteristics similar to those identified in clinical education. It is now known that feedback can impact a range of factors for software engineers, and can have an impact that lasts at least until the end of the work day. The importance of the different characteristics of feedback remains unclear. How does each characteristic help to mould the impact? Are there important feedback characteristics? Are there less important feedback characteristics? And does this vary for each person?

The next study will specifically investigate the characteristics of feedback, and investigate the affect that the values of feedback characteristics have on the impact of received feedback. This study has provided powerful insight and direction for the future study, but remains indicative and falls short of providing key findings surrounding the impact of the characteristics of feedback.

An updated map of the research progress is shown in Figure 5.31.



Research Stage:	Literature Review	Feedback Scope	Feedback in SE
Goal:	Identify previous findings from research investigating the impact of feedback in software engineering environments. Find theories of motivation that identify feedback as a factor in motivation and identify their use and definition of feedback.	Investigate the definition of feedback in software engineering environments and comparing findings with clinical education. Investigate if software practitioners report receiving feedback in their development environment.	Investigate how feedback typically occurs in software development environments. Investigate the overall impact that feedback can have on software practitioners.
Outcomes:	Feedback is a factor affecting the motivation of software engineers. Research has argued that feedback is the most important need of IS professionals. Feedback is a factor in four theories of motivation. Theories of motivation use feedback to provide the recipient with the knowledge of the outcome of their actions.	Feedback frequently occurs in software engineering environments. Feedback is identified as having characteristics comparable to those identified in another discipline. Four aspects were identified as being affected by received feedback.	Feedback is the most common method for software practitioners to assess their progress on a project. Feedback is typically received directly from a colleague or manager. Feedback can affect software practitioners' work, their attitude their behaviour, their motivation, their productivity, their job satisfaction, their enjoyment, and their feelings. Feedback often affects the feelings of software practitioners, with every feedback instance typically provoking multiple feelings. Different forms of feedback can have a different impact on the identified aspects.

Research Stage Key:

Literature - Literature Review, chapter 2
Feedback Scope - Feedback Scoping Study, chapter 3
Feedback in SE - Feedback in One Software Engineering Environment, chapter 5

FIGURE 5.31 – UPDATED RESEARCH PROGRESS MAP

CHAPTER 6

THE SHORT-TERM IMPACT OF RECEIVED FEEDBACK

The study presented in this chapter investigated the impact that instances of feedback were reported to have on the motivation, behaviour, attitude, productivity and job satisfaction of software engineers. This study builds on the findings of the Feedback Scoping Study (chapter 3) and Feedback in One Software Engineering Environment Study (chapter 5) by narrowing the scope of the study based on the findings of the previous studies. The previous studies found that software engineers discuss feedback as the combination of ten characteristics, and that the values of feedback characteristics can change the impact of feedback. This study focuses on investigating the relationship between feedback reported as occurring in software engineering environments and the aspects identified as being affected by received feedback, enhancing the understanding of the relationship between feedback characteristics and feedback affect.

The following sections discuss, identify and explain the research design for this study (6.1), present the participants (6.2), discuss the analysis approach taken (6.3), present the findings from this study (6.4), discuss the threats to validity (6.5), discuss the findings from this study (6.6), and finally presents a state of play updated with the knowledge gained from the findings of this study (6.7).

6.1 RESEARCH DESIGN

The identified gap in our current knowledge is the relationship between the values of the characteristics of feedback and the impact they have on the aspects reported as being affected by feedback. To investigate this area it was necessary to elicit the thoughts and perceptions of active software engineers.

To ensure the research design is appropriate for the participants, data was collected in this study from engineers working in the same environment as those in the Feedback in One Software Engineering Environment Study (chapter 5). This allowed the use of earlier findings to influence the research approach taken during this study.

Using the data collected during the Feedback in One Software Engineering Environment Study (chapter 5) several scenarios were created, portraying feedback similar to those that had been reported during the previous study (chapter 5). These were used to investigate the impact of a change in value of a feedback characteristic. The scenarios were created to be representative of different forms of feedback reported in the previous study (chapter 5), allowing this study to investigate the effect of the values of feedback characteristics on different aspects of software engineers.

To make the scenarios suitable for all participants, phrases commonly found in the environment such as ‘project manager’ and ‘software tester’ were used.

TABLE 6.1 – FEEDBACK CHARACTERISTIC VALUES FOR SEVEN SCENARIOS

CHARACTERISTIC	SCENARIO 1	SCENARIO 2	SCENARIO 3	SCENARIO 4	SCENARIO 5	SCENARIO 6	SCENARIO 7
Type:	Positive	Negative	Positive	Negative	Positive	Negative	Positive
Timeliness:	Instant	Delayed	Delayed	Delayed	Delayed	Delayed	Delayed
Medium:	Face to face	Face to face	Email	Face to face	Overheard	Face to face	Face to face
Source:	Peer	Project Manager	Division Head	Peer Through PM	Peer	Peer	Project Manager

CHARACTERISTIC	SCENARIO 1	SCENARIO 2	SCENARIO 3	SCENARIO 4	SCENARIO 5	SCENARIO 6	SCENARIO 7
Subject:	Help with problem	Project Progress	Performance	Problem with Work	Help with problem	Breaking build	Recent Performance
Setting:	Casual	Stand-up Meeting	Electronic	One-to-one Meeting	Casual	Casual	One-to-one Meeting

Each scenario had a different set of values for the feedback characteristics (Table 6.1). To aid understanding and clarity, and to engage the participants with the questions, six fictitious people were created with roles representative of people already found in their environment. These people were used as the source of the feedback during the scenarios. For each scenario, the engineers were asked if the scenario would have an impact on any of the listed aspects (attitude, behaviour, motivation, productivity and job satisfaction).The aspects used in this study were identified in the literature review and reported as being affected by received feedback in the previous study (chapter 5).

6.1.1 RESEARCH CONDUCT

The participants were emailed a copy of the scenario form (appendix C section 1) which included an introduction to the research, followed by the definition of the six fictitious people. After the introduction, the participants were presented with the seven feedback scenarios. The participants were asked to read through the form, print it out, and indicate if each scenario would impact any of the five listed aspects. Individual meetings were arranged with each participant to discuss their responses. Each meeting was audio recorded to aid future data analysis.

During each meeting, the participants were asked to discuss the reasoning behind their response for each scenario. Each participant was asked to respond to changes in each scenario, for example the interviewer would change the source of the feedback from Tom the project manager, to Rick a software engineer, and ask the participant if this change would alter their response about the impact of the scenario on any of the five listed aspects. Some of the participants arrived to the interviews without already completing the scenario form. They completed the form before the interview began. The interviews lasted between 10 and 30 minutes.

6.1.2 PILOT STUDY

The discussed research design was piloted to ensure that the questions that were asked were suitable, and that the data collected was appropriate for the intention of the study. From the findings of the pilot study, the form presented to the participants was altered slightly to make it more aesthetically acceptable and to improve the ease of navigating between questions and interpreting the format of the scenarios correctly. Minor changes were made to the definition of the six fictitious people, and terms used to describe their job role were informed from the findings discussed in the Feedback in One Software Engineering Environment Study (chapter 5) , for example 'project manager', 'software tester', and 'division head' were all found to be common terms used at Red Gate.

6.1.3 ETHICAL CLEARANCE

Ethical clearance had already been obtained from the Open University's Human Research Ethics Committee for the research carried out at Red Gate (appendix B section 3). A copy of the consent form presented to the participants is also included in appendix B section 4.

6.2 PARTICIPANTS

Data was collected from 16 software engineers at Red Gate (see section 5.2 for a description of the Red Gate environment). 12 of the engineers had also been participants in the Feedback in One Software Engineering Environment Study (chapter 5). The software engineers had the same demographics as those described in section 5.3: they all had a minimum of 6 months experience as a software engineer, and had been employees at Red Gate for between six weeks and six years.

6.3 DATA ANALYSIS

Two forms of data were collected during this study.

1. The participants each printed a copy of the document containing the scenarios, completing and returning it to the interviewer during the meeting. This response data was extracted from each scenario document, tabulated, and descriptive statistics identified. The electronic copies of the tabulated data were stored in a secure password-protected folder on a password-protected computer. The original copies of the scenarios were shredded and recycled to ensure that the participants' confidentiality was maintained.

2. The second form of data was the recording of the interview with each participant. The audio was recorded using a handheld device. The recordings were downloaded onto a password-protected computer and then erased from the original recording device to ensure that the recordings were securely stored. The electronic audio recordings were stored in a secure password-protected folder on a password-protected computer. The audio recordings for each of the interviews were transcribed.

The transcripts were divided into segments, and each response was coded in relation to the scenario or change in scenario that was being discussed. The analysis was data-driven, and was divided into two distinct parts: Initially, numerical details of the data were identified to investigate emergent patterns. Next, an inductive analysis of the transcribed responses was conducted scenario-by-scenario, and emergent themes were identified that related to the impact of the feedback strictly on the basis of what was evident in the data.

6.4 FINDINGS

During this study software engineers responded to example scenarios describing when they would receive feedback. The engineers reported whether they believed any of the 5 listed aspects would be affected if they were to actually receive the feedback in their development environment. During the follow-up interview, participants were asked to expand on their responses and indicate if they believed their response would change if the presented scenario changed, for example the source of the feedback changed from a project manager to a software engineer.

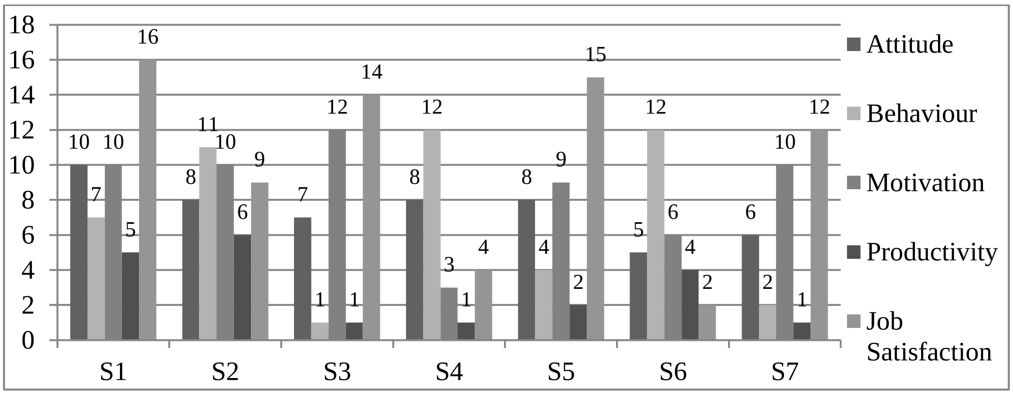


FIGURE 6.1– AFFECTED ASPECTS IN 7 SCENARIOS FOR 16 SOFTWARE ENGINEERS

Figure 6.1 shows the compiled indicated responses to each scenario. Table 6.1 gives definitions of the feedback characteristics present in each scenario, and the full scenarios presented to the participant can be seen in appendix C section 1. What becomes clearer in Figure 6.2 is that the form of feedback, specifically the polarity (positive or negative) of the feedback content, has an effect on the number of engineers reporting it as affecting some of the aspects.

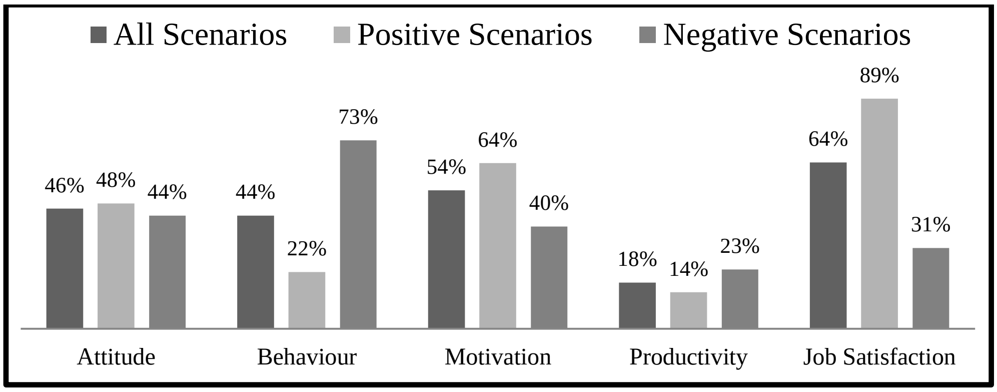


FIGURE 6.2 – AFFECTED ASPECTS FOR ALL, POSITIVE, AND NEGATIVE FEEDBACK SCENARIOS

Additional images of the data analysis showing the participants responses to positive, negative, and all scenarios divided by impact can be seen in appendix C section 2.

In scenario 1, all of the engineers reported that the scenario would have an impact on their job satisfaction. All of the scenarios would have an impact on each of the five aspects for at least one participant.

Overall, as can be seen in Figure 6.2, the scenarios had an impact on the engineers' attitude in 46% of all scenarios, behaviour in 44% of all scenarios, motivation in 54% of all scenarios, productivity in 18% of all scenarios, and job satisfaction in 64% of all scenarios.

Across all of the scenarios, attitude was affected at least once for 13/16 engineers, behaviour was affected at least once for all 16/16 engineers, motivation was affected at least once for all 16/16 engineers, productivity was impacted at least once for 10/16 engineers, and job satisfaction was impacted at least once for all 16/16 engineers.

6.4.1 SCENARIO ANALYSIS

This section presents the results of the analysis of the data collected during the interviews with the software engineers. The participants were asked to expand on their pre-filled response to each scenario. From the participant's expanded response to each scenario, emergent themes were identified. Each emergent theme discussed by two or more participants is reported in this section. Each participant could discuss multiple themes. Not every additional scenario permutation was addressed with every participant because of time constraints. Scenario permutations were prioritised to ensure as many software engineers as possible responded to the same scenario permutations. For each scenario permutation, the number of participants elicited to discuss the effect of the scenario permutation is indicated in brackets next to each identified theme. All of the themes presented in this section emerged from an inductive analysis of the participant's responses to an individual scenario of when they would receive feedback.

Scenario 1 – "You're working with Rick (Software Engineer) on a piece of code he's having trouble with. After helping him, he thanks you and tells you what a life saver you are." The participants discussed that the feedback in this scenario would:

- *Increase their willingness to help* (7/16) "it would probably make me more prone to helping people".
- *Give them the knowledge that they were being appreciated* (7/16) "your work at the company is appreciated".
- *Make them feel like they were doing a good job* (5/16) "feel like I've done a good job".
- *Make them enjoy helping people* (3/16) "enjoy sort of helping other developers".

If the feedback provided in scenario 1 were sent by email, the participants reported that it would have:

- No change (5/16) because:
    - It's normal: "it's just one of the many different ways of communicating in the office."
    - Of a personal preference: "some people communicate better by email, some people communicate better verbally".
- Less impact (4/16) because:
    - "You lose a lot in an email".
    - They would "always take away more from face to face communication".
- More impact (3/16) because:

    - If "they've gone to the effort of writing an email that's sort of saying something more pronounced" and "that's even better".
- Prefer face to face (3/16) because:
    - Establishing the reason for the feedback is easier "get a sense of whether they're actually being genuine instead of just following process".

If the feedback provided in scenario 1 were from their project manager, the participants reported that it would have:

- No change (7/16) because:
    - It's the appreciation that's important "as long as someone appreciates the work you're putting into it and the work that you're doing".
- Less impact (4/16) because:
    - Of their position "I am more interested in my equals than I am project managers".
    - Of their relationship with the source "it's a little more removed if it's come from the project manager, it's a bit more kind of process oriented."
- More impact (2/16) because:
    - It may increase reputation and exposure "I'd actually be a little bit more pleased because then it kind of ties into stuff like reputation and things like that, like all of a sudden you're perceived in a wider sphere as being a positive influence."

Scenario 2 – "At a stand up meeting, Tom (Project Manager) tells the team how he's disappointed with the current progress on part of the project. You feel responsible for this lack of progress." The participants discussed that the feedback in this scenario would:

- *Make them want to fix the problem* (5/16) "fix whatever's wrong with the project".
- *Have a negative impact on their feelings* (4/16) "I'd just feel really bummed."
- *Motivate them to alter their behaviour* (3/16) "I'd feel like I'd want to change".

- *Make them work harder* (3/16) "work a bit harder on whatever it is that's been holding up the project."

If the feedback provided in scenario 2 came from the division head, the participants reported that it would have:

- No change (6/15).
- More impact (4/15) because:
    - It would be perceived as a bigger problem "it would change behaviour to a bigger extent as I would see this as a bigger problem".
- Less impact (2/15) because:
    - Of the feedback lacking validity "it would be less valuable as feedback because I feel that the division head doesn't really know what's happening in the project".
    - Of their expectations being unrealistic "you'd feel that they might just be being unrealistic with their expectations".
- The impact would depend (2/15) because:
    - Of why he felt disappointed "depends why Boris thinks we should have progressed further".

Scenario 3 – "You receive an email from Boris (Division Head) telling you what a brilliant job you've done lately and how he's impressed with your performance." The participants discussed that the feedback in this scenario would:

- *Make them want to do more of the same* (5/16) "you'd want to carry on doing the work you're doing".
- *Have a positive impact on their feelings* (4/15) "you're delighted if someone has genuinely recognised that you've done a good job."
- *Make them feel appreciated* (3/15) "knowing that people are appreciating the work that I'm doing".

If the feedback provided in scenario 3 came from the project manager, the participants reported that it would have:

- No change (6/15).
- More impact (4/15) because:

  - Because they're closer to the recipient "because they are part of the team" and "more closely involved with what I do on a day to day basis so I'd value his feedback more."
  - Because it would be more believable, as the project manager would be able to "comment validly" and their comments are "more believable".
- Less impact (3/15) because:
  - It's expected "it's my understanding that project managers, they tend to tell you that you're doing a good job quite regularly anyway".

If the feedback provided in scenario 3 came from the CEO, the participants reported that it would have:

- Increased the impact (11/15) because:
  - Of the feedback would be less frequent because you "interact with the CEOs [less frequently] so if something does come from them it's quite nice."
  - Of the source's job title "higher up the structure [the feedback source is] the more impact [feedback] has."
- No change (3/15).

Scenario 4 – "During a one to one meeting with Tom (Project Manager), he talks to you about a problem with your work on a recent project. You suspect Tom (Project Manager) is relaying feedback from Rick (Software Engineer)." The participants discussed that the feedback in this scenario would:

- *Make them want to fix the problem* (10/16) "I'd try to fix whatever was the problem".
- *Change their behaviour* (7/16) "I would try and change my behaviour towards Tom or towards Rick".

Scenario 5 – "Gary (Software Engineer) asks you to help him with a problem he's stuck on. After you help him he thanks you, and you overhear him telling Simon (Software Tester) what a great help you've been." The participants discussed that the feedback in this scenario would:

- *Have a positive impact on their feelings* (7/16) "I would feel quite pleased that I'd helped."

- *Let them know they're appreciated* (5/16) "it's good to know that you're appreciated."
- *Make them enjoy working* here (4/16) "I'd enjoy my job more if I heard this."
- *Make them feel motivated to help people* (3/16) "spend more time helping people."

If the feedback provided in scenario 5 were overheard being told to the project manager, the participants reported that it would have:

- No change (10/15).
- Increase the impact (2/15) because:
  - The boss would be aware of what you're doing "your boss knows that you're doing useful stuff."
- Would be preferred (2/15) because:
  - It would increase awareness of contributions "visibility on your contribution."

Scenario 6 – "Simon (Software Tester) comes over to speak to you. He has some bad news – recent changes you made broke the system." The participants discussed that the feedback in this scenario would:

- *Change their behaviour* (5/16) "change your behaviour accordingly".
- *Make them want to fix the problem* (6/16) "hunker down and fix it".
- *Have a negative impact on their feelings* (5/16) "feel dreadful that I've broken something".

If the feedback provided in scenario 6 were sent by email, the participants reported that it would have:

- No change (8/14).
- Depends on the content and wording of the actual email (3/14) because:
  - Of the phrasing "vary massively on how it's phrased".

Scenario 7 – "During a one to one meeting with Tom (Project Manager), he tells you how happy he is with your recent performance." The participants discussed that the feedback in this scenario would:

- *Have a positive impact on their feelings* (7/16) "it feels good to get good feedback".

- *Make them want to continue doing the same work* (6/16) “continue doing the same sort of work that you’d been doing.”
- *That the recognition is satisfying* (5/16) “personal recognition is what is most satisfying.”

6.4.2 SUMMARY OF FINDINGS

The data showed that both positive and negative feedback can have an effect on the software engineers. Positive feedback most often led to a positive impact on job satisfaction and motivation, and negative feedback most often led to an impact on behaviour. Productivity was affected less than the other aspects, with attitude being affected for a similar number of engineers across all feedback scenarios.

The hierarchical position of the person providing the feedback had a varied impact on the effect of the feedback. The responses to a change in hierarchical position were the most divided of all responses from the engineers. Some engineers preferred feedback from their colleagues, because they felt that their colleagues were “in the trenches” and knew more of the “day to day” details of their work. Other engineers preferred the feedback from their project manager, as he was considered “the boss”, and they believed it was good to be seen to be doing good work by their “boss”.

A number of software engineers preferred feedback from their division head to feedback from their project manager. They felt that their project manager “always tells me he’s happy” so they “just get used to being told this every time”. However some software engineers felt that their division head was too far removed to know the details of their work, and instead preferred feedback from their project manager who knew more details of their work.

The difference in the impact of the hierarchical position may be explained by *technical respect*. One software engineer only cared for the opinion of someone with whom he had previously worked and whom he knew was a capable software engineer, not necessarily a capable manager. Other software engineers shared a similar view: the opinion of their colleagues mattered more than that of someone who they knew as a hierarchical superior. In contrast, other software engineers were more interested in feedback from someone above them in the hierarchical chain, such as the division head who they didn't already receive consistent feedback from.

Most of the software engineers enjoyed receiving feedback from their CEO. The engineers didn't expect the CEO to know specific details about their work, but they consistently reported that being recognised by the CEO was a positive event. This provided an interesting outcome from the analysis, as both the CEO and the division head were not expected to know precise details of completed work, but the CEO's feedback was always received positively while feedback from the division head had varying value across the engineers.

The impact of feedback coming via email rather than during a face to face conversation was varied. A change in the medium used to provide the feedback was most often reported as not changing the impact of the feedback, however in response to scenario one being received by email, four engineers reported that it would increase the impact, and three engineers reported that it would reduce the impact. The engineers seemed more interested in the content of the email, and not the medium that was being used. However some engineers did state that they preferred face-to-face feedback. The impact of the medium is best demonstrated in scenario 3, where positive feedback is received by email, and is reported as having an impact on the majority of the engineer's motivation and job satisfaction. The findings from this study suggest that the medium is considered a preference for software engineers, and has less effect on the impact of received feedback than the source feedback characteristic and the content polarity.

The data collected in this study suggests that these software engineers do not connect motivation with productivity. Positive feedback affected motivation in 41/64 (64%) instances, yet in these instances productivity was only reported as being affected 20% (8/41) of the time. Negative feedback impacted motivation 19/48 (40%) but in these same instances productivity was reported as being affected 47% (9/19) of the time.

6.5 THREATS TO VALIDITY

The findings of this study are subject to three main threats to validity. The first and second threats to validity are the same threats to validity as presented in Chapter 5, section 5.6: the external threat to validity due to the sampling approach and the number of participants.

In addition to the first two threats to validity, as 12 of the participants had already completed the previous study, and this study builds on the findings of the previous study, there is a threat of ‘priming’ from the previous study, possibly causing bias in their responses in this study. The participants may have responded to scenarios of feedback in the same way as they reported their reactions to feedback in the previous study, rather than reflecting on and responding to each scenario of feedback during this study. This provides a threat to the external validity of the findings. This threat was limited by the design of the study, which provided scenarios of feedback that were abstract and did not clearly identify any instance of feedback reported by the participants in the previous study.

6.6 SUMMARY

The findings from this study shed further light on the impact of feedback, and provide evidence investigating the relationship between instances of feedback and their impact on five aspects (attitude, behaviour, motivation, productivity and job satisfaction). The indications from these findings are that positive feedback increases job satisfaction, negative feedback leads to a change in behaviour, and that the medium and source feedback characteristics are a preference that can alter the impact of feedback.

These findings increase knowledge and understanding of feedback in software engineering, but provide more questions. The next study investigates these findings with more software engineers working in different software development environments to ensure that the findings of this study are not specific to the Red Gate environment, and to provide evidence that supports or disputes the findings of this study.

6.7 SO WHAT DO WE KNOW NOW?

The data collected during this study offers some initial insights into the impact the different values of feedback characteristics may have, and suggests that theories of motivation (discussed in chapter 2) may not accurately represent feedback and its affect in software engineering environments.

The results of this study suggest there is a difference between the impact positive and negative feedback can have on motivation and job satisfaction. Both job satisfaction and motivation were impacted across more participants by positive feedback (JS: 89%, M: 64%), than by negative feedback (JS: 31%, M: 40%), even though the negative feedback was still providing the recipient with the knowledge of the result of their actions. Three theories of motivation (Job Characteristics Theory, Achievement Theory, Goal Setting Theory) identify feedback as a factor affecting motivation by providing individuals with the knowledge of the result of their actions. This finding suggests that in software engineering environments the effect of feedback that provides individuals with the knowledge of the result of their actions can vary depending on the values of the feedback's characteristics.

Negative feedback was reported as having an impact on behaviour in 73% of all occurrences. The software engineers reported that they wanted to fix the problem when they received negative feedback, and that if required, they would change their behaviour to achieve this.

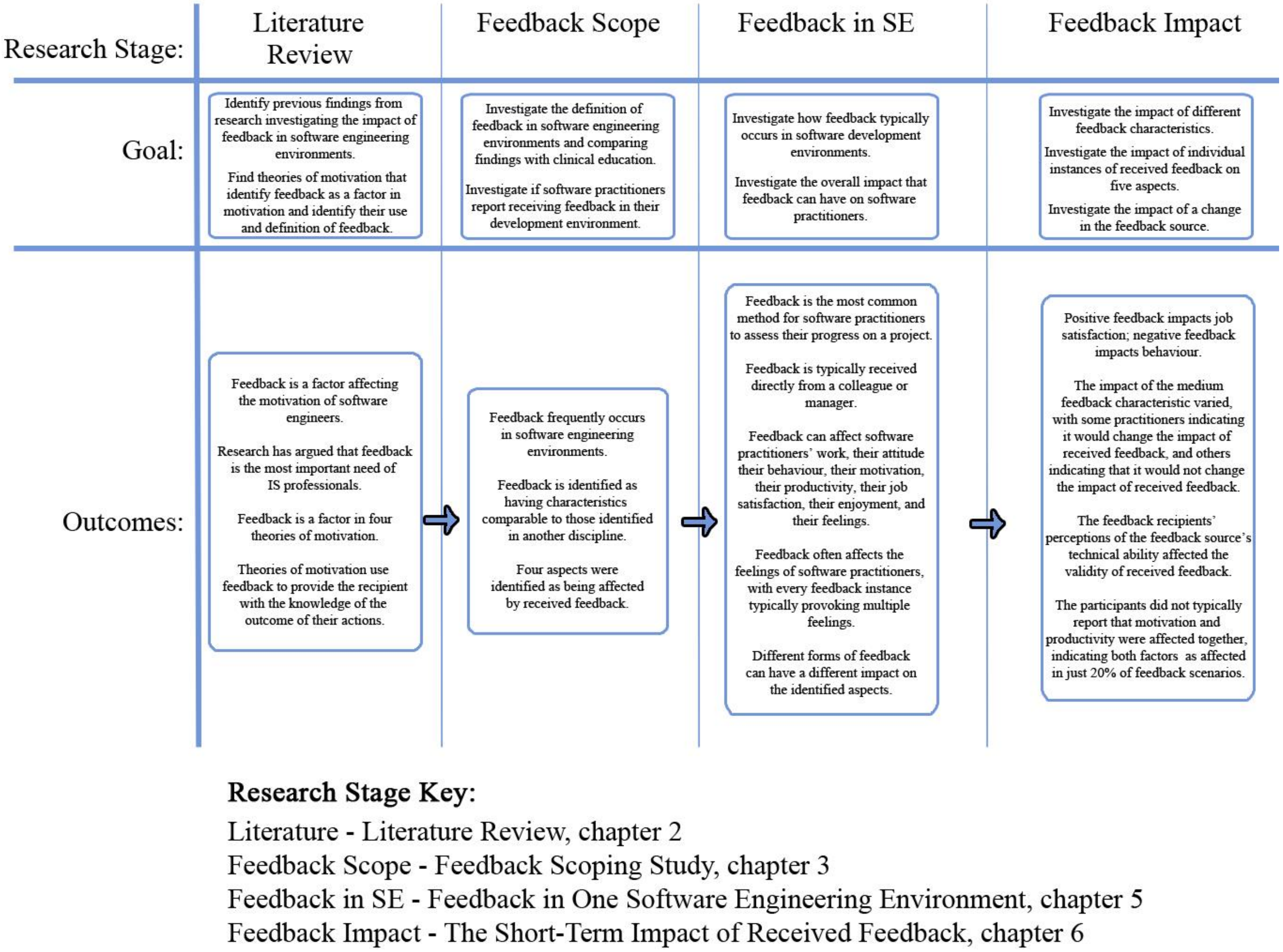
The impact of the medium used to distribute the feedback varied from no impact, an increase in the impact of the feedback, and a reduction in the impact of the feedback. The medium did not appear to be a crucial factor, and while the data provided mixed results on how it would change the impact of received feedback, the software engineers were more concerned with the actual content of the feedback and who it was from, rather than how they received it, suggesting that the feedback medium is a preference and is less influential than other feedback characteristics.

Technical respect for the source of the feedback was influential. Regardless of hierarchical position, technical respect was found to influence the validity of feedback received and the impact it had on the participants. Receiving feedback from someone who they did not have technical respect for often had minimal impact, if technical respect was valued by the feedback recipient.

The participants only linked productivity with motivation in 20% of the positive feedback scenarios. This suggests that these software engineers themselves do not recognise a relationship identified in previous literature (Beecham *et al.*, 2008) suggesting that an increase in motivation increases productivity.

The gap in knowledge identified during the Feedback in One Software Engineering Environment Study (chapter 5) has been investigated during this study, however the same limitations from the Feedback in One Software Engineering Environment Study (chapter 5) are present in this study. The participants all work at the same company, and all work in the same environment. The data may be altered by the environment that the software engineers work in, their company culture, and their development methodology. The small number of participants also hinders the representativeness of the findings.

An updated map of the research progress is shown in Figure 6.3.



Research Stage:	Literature Review	Feedback Scope	Feedback in SE	Feedback Impact
Goal:	Identify previous findings from research investigating the impact of feedback in software engineering environments. Find theories of motivation that identify feedback as a factor in motivation and identify their use and definition of feedback.	Investigate the definition of feedback in software engineering environments and comparing findings with clinical education. Investigate if software practitioners report receiving feedback in their development environment.	Investigate how feedback typically occurs in software development environments. Investigate the overall impact that feedback can have on software practitioners.	Investigate the impact of different feedback characteristics. Investigate the impact of individual instances of received feedback on five aspects. Investigate the impact of a change in the feedback source.
Outcomes:	Feedback is a factor affecting the motivation of software engineers. Research has argued that feedback is the most important need of IS professionals. Feedback is a factor in four theories of motivation. Theories of motivation use feedback to provide the recipient with the knowledge of the outcome of their actions.	Feedback frequently occurs in software engineering environments. Feedback is identified as having characteristics comparable to those identified in another discipline. Four aspects were identified as being affected by received feedback.	Feedback is the most common method for software practitioners to assess their progress on a project. Feedback is typically received directly from a colleague or manager. Feedback can affect software practitioners' work, their attitude their behaviour, their motivation, their productivity, their job satisfaction, their enjoyment, and their feelings. Feedback often affects the feelings of software practitioners, with every feedback instance typically provoking multiple feelings. Different forms of feedback can have a different impact on the identified aspects.	Positive feedback impacts job satisfaction; negative feedback impacts behaviour. The impact of the medium feedback characteristic varied, with some practitioners indicating it would change the impact of received feedback, and others indicating that it would not change the impact of received feedback. The feedback recipients' perceptions of the feedback source's technical ability affected the validity of received feedback. The participants did not typically report that motivation and productivity were affected together, indicating both factors as affected in just 20% of feedback scenarios.

Research Stage Key:

Literature - Literature Review, chapter 2
Feedback Scope - Feedback Scoping Study, chapter 3
Feedback in SE - Feedback in One Software Engineering Environment, chapter 5
Feedback Impact - The Short-Term Impact of Received Feedback, chapter 6

FIGURE 6.3 – UPDATED RESEARCH PROGRESS MAP

CHAPTER 7

THE EFFECT OF 'SOURCE' AND 'MEDIUM' FEEDBACK CHARACTERISTICS

This chapter builds on the findings of the Feedback Scoping Study (chapter 3), the Feedback in One Software Engineering Environment Study (chapter 5), and the Short-Term Impact of Received Feedback Study (chapter 6). The previous studies identified feedback characteristics, identified the impact of the values of feedback characteristics, and identified two characteristics that were found to have the clearest effect on the impact of received feedback. This study builds on these findings by further investigating the effect of the source and medium feedback characteristics with a larger group of software engineers, working in other companies, working in different software development environments, and working in different management structures. This study builds on the findings of the previous studies presented in chapters 3, 5 and 6, focusing on the effect of the two characteristics found to have the most influence on the impact of received feedback (source and medium), and collects empirical data from over 150 people who reported themselves as being active software engineers.

The following sections discuss, identify and explain the research design for this study (7.1), present the participants (7.2), discuss the analysis approach taken (7.3), present the results (7.4), discuss the threats to validity (7.5), discuss the findings from this study (7.6), and finally presents a state of play updated with the knowledge gained from the findings of this study (7.7).

7.1 RESEARCH DESIGN

The focus of this study was to identify the impact of a change in value for two feedback characteristics: medium and source. The investigation sought to elicit responses from a wider population of software engineers who work in different environments to those identified during the Feedback in One Software Engineering Environment (chapter 5) and The Short-Term Impact of Received Feedback (chapter 6) studies.

An online survey was designed and completed with active software engineers who considered themselves first and foremost software engineers, who worked in team environments, and reported that they often wrote new code. The use of the online survey enables the collection of data from multiple participants without requiring direct contact. The survey can also be distributed easily to wider audiences, and is less time-consuming and intrusive than a direct interview. The survey participants were asked to respond to receiving different forms of feedback. In the following section the design of the survey is discussed in detail. The online survey was designed with the goal of collecting data investigating the focus area in roughly ten minutes to help encourage a larger number of participants.

7.1.1 SURVEY DESIGN

According to Fink (2002: 165), “surveys are systems for collecting information from or about people to describe, compare, or explain their knowledge, attitudes, and behaviour." Robson defines a survey as a "collection of standardized information from a specific population, or some sample from one, usually, but not necessarily by means of a questionnaire or interview" (Robson, 2002: 185). Oppenheim states that a self-administered questionnaire can ensure "a high response rate, accurate sampling and a minimum of interviewer bias" (Oppenheim, 2000: 103). This study is experimental, characterised by Fink as "the comparison of two or more groups, at least one of which is experimental and the others of which are control (or comparison) groups" (2002: 161). This survey uses sub-groups within the sampled population to investigate the representativeness of findings identified in the earlier empirical studies.

This study investigates the effect of the values of feedback characteristics, specifically the source and the medium, by collecting information from software engineers about how they believe they would react to different instances of feedback.

The designed experimental survey is a cross-sectional survey, collecting data at one point in time for a self-selected sample of software engineers. The design of the survey ensured that "the words in questions…all respondents understand their meaning" (Fowler and Floyd, 1995: 85) and that the text included in the survey included words and terms which were unlikely to be ambiguous, a design practice encouraged by Fowler and Floyd: "words or terms must be used that have meanings that are likely not to be shared" (Fowler and Floyd, 1995: 85).

Vaus (1991: 83-86) recommends several good practices for survey design and these were followed:

- Remove ambiguity.
- Avoid direct questions on sensitive topics (in interview situations).
- Ensure question's frame of reference is clear.
- Avoid creating opinions.
- Use personal wording if you want the respondents' own feelings etc.
- Avoid unnecessary or objectionable detail.
- Avoid prior alternatives.
- Avoid producing response sets.

Robson (2002: 190) extends the good practices for designing surveys and suggests that there are three base factors in securing a good response rate to a remote questionnaire:

- Appearance of the questionnaire.
- Clarity of wording and simplicity of the design.
- Arrangement of contents to maximize co-operation.

Oppenheim (2000: 104) recommends the following steps to increase response rates:

- Providing advanced warning.
- Explaining how the respondent came to be chosen.
- Gaining sponsorship by someone expected to be influential to the respondents.
- Providing incentives for participation
- Treating data as confidential

When designing the survey the appearance, clarity of wording, and arrangement of contents was carefully considered to improve the response rate of the survey. Participants were invited, their selection explained, sponsorship by influential parties sought, and assurances of confidentiality provided. Further details on the information provided to the participants and the process in which the participants were contacted can be found in section 7.2.

The survey adhered to 5 further elements of good practice when designing online surveys:

1. The online survey was designed and tested to support multiple platforms and browsers (Yun and Trumbo, 2000: 29).
2. Multiple submissions from the same user were limited (Keheo and Pitkow, 1996: 42).
3. Participant’s answers were saved at multiple occasions (Smith, 1997: 45).
4. Participants were provided with the ability to provide both closed and open-ended responses to questions (Yun and Trumbo, 2000: 33).
5. Immediate “thank you” feedback was provided upon the completion of the survey (Smith, 1997:39).

The online survey was structured as shown in table Table 7.1.

TABLE 7.1 - ONLINE SURVEY STRUCTURE

Page 1:	Introduction to the research, the topic of the survey, the format and questions included in the survey, and contact details should the participant have any further questions.
Page 2:	Collection of demographic information.
Pages 3-14:	Collection of responses to twelve questions and scenarios.
Page 15:	Open comments box for any additional comments from the respondents. Request for the participants to 'opt-in' for a follow-up interview, and collection of contact details and preferences when relevant.
Page 16:	'Thank you' page, end of the online survey.

The demographic questions, the possible responses to each question, and the type of response are shown in Table 7.2, and a screenshot of the online survey page as seen by the participants can be seen in appendix D section 1. Please note, if a participant selected 'other', a free-text box would appear directly below the selected question for the participant to enter an alternative response.

TABLE 7.2 – DEMOGRAPHIC QUESTIONS ASKED DURING SURVEY

Question	Response Options	Response Type
What is the highest level you are currently educated to?	• GCSE • A-level • Bachelor's Degree • Master's Degree • Doctorate Degree	Single-choice Radio Buttons
Approximately, how many years' experience do you have developing software?	• Less than 6 months • 6 months • 1 year • 2 years…29 years • 30+ years	Single-choice Select-box List
Approximately, how long have you been at your current employer?	• Less than 6 months • 6 months • 1 year • 2 years…29 years • 30+ years	Single-choice Select-box List
Which of the following types of development methods have you had experience with?	• Agile Methods • Waterfall Model • Spiral Model • Iterative Development • Other ○ ('Other' selection opens free-text box)	Multiple-choice Checkboxes
Which of the following tasks are typical of your current job?	• Requirements Gathering • Systems Design, Writing Code • Systems Testing • System Maintenance • Other ○ ('Other' selection opens free-text box)	Multiple-choice Checkboxes

Question	Response Options	Response Type
Including software engineers and other colleagues, what size development team are you typically in?	• I work on my own • 2-3 people • 4-6 people • 7-9 people • 10-15 people • 16-30 people • 31+ people	Single-choice Select-box List
How many other software engineers are typically in your development team?	• No other software engineers • 1 other software engineering • 2 other software engineers…. 29 other software engineers • 30+ other software engineers	Single-choice Select-box List

In the following sections the questions and scenarios included in the online survey during pages 3-14 are presented and discussed.

TABLE 7.3 – FEEDBACK CHARACTERISTICS FOR EACH SURVEY QUESTION

CHARACTERISTIC	SCENARIO 1	SCENARIO 2	SCENARIO 3	QUESTION 4	QUESTION 5	QUESTION 6
Type:	Positive	Positive	Positive	Not applicable – a rating of the value of feedback from multiple sources	Not applicable – a rating of the impact praise would have from multiple sources	Not applicable – a rating of the impact critical comments would have from multiple sources
Timeliness:	Delayed	Delayed	Delayed			
Source:	Manager	Software Engineer	Senior Manager			
Subject:	Good job on recent work	Recently wrote good bit of code	Good job on a recent project			
CHARACTERISTIC	SCENARIO 7	SCENARIO 8	SCENARIO 9	SCENARIO 10	SCENARIO 11	SCENARIO 12
Type:	Positive	Positive	Positive	Negative	Negative	Negative
Timeliness:	Delayed	Delayed	Delayed	Delayed	Delayed	Delayed
Source:	Manager	Software Engineer	Senior Manager	Manager	Software Engineer	Senior Manager
Subject:	Good job on recent work	Good bit of code	Good job on a recent project	Recently did some poor work	Recently wrote poor bit of code	Poor job on a recent project

The participants were asked to respond to a combination of 12 scenarios and questions. The relevant feedback characteristics for each scenario are shown in Table 7.3. Questions 4, 5, and 6 do not have feedback characteristics, as they are asking the respondents to rate the value and impact of feedback, praise, and critical comments received from six different sources (section 7.1.2). Overall, four different types of questions and scenarios were presented to the participants:

Scenario Type 1 – Scenarios 1-3.

- The participants were presented with an instance of feedback and asked to indicate how they would prefer to receive the feedback by selecting one or several of the listed media.
- The participants were then asked if receiving this feedback from any other media would change the impact of the feedback.
- If the participant indicated that receiving the feedback from any other media would change the impact of the feedback, the participants were asked to indicate if the change in impact would be minor, moderate, or major.

Question Type 1 – Questions 4-6.

- The participants were asked to indicate the value or impact of receiving feedback, praise, and critical comments from six listed sources of feedback by selecting a value on a 5-point Likert scale.

Scenario Type 2 – Scenarios 7-9.

- The participants were presented with an instance of feedback and asked to indicate if it would have a positive impact on any of 6 listed aspects by selecting one or multiple aspects.

Scenario Type 3 – Scenarios 10-12.

- The participants were presented with an instance of feedback and asked to indicate if it would have a negative impact on any of 6 listed aspects by selecting one or multiple aspects.
- The participants were then asked how they would react after receiving this feedback by selecting one or several of the 4 listed options.
- The participants were given a 5th option of selecting 'other', and indicating a different reaction through a free-text box.

After completing the 12 questions and scenarios, the participants were presented with the penultimate page with an additional comment box, and after the participants completed that page, a thank you message was displayed and their data marked as completed.

In the following section each question and scenario is presented in detail.

7.1.2 SURVEY COMPOSITION

The following paragraphs present the feedback displayed in the different scenarios and questions, and the lists the questions posed to the participants in each scenario or question. A page-by-page replica of the online survey can be seen in appendix D section 1. The scenarios and questions fall into four groups, as outlined in Table 7.1. Each group of scenarios or questions is enclosed within a box, and preceding the response options for that group of scenarios or questions.

Scenario 1 – *Your line manager tells you he thought you did a good job on some recent work...*

Scenario 2 – *A fellow software engineer tells you he thought you recently wrote a really good bit of code...*

Scenario 3 – *A senior manager tells you he thought you did a good job on a recent project...*

Questions presented to the participants in Scenarios 1, 2 and 3:

How would you prefer to receive this feedback? Select all that apply

Checkbox Options:

- *Team Meeting.*
- *Casual Chat.*
- *Email.*
- *One-to-one Meeting.*
- *Notice Board.*

Would receiving this feedback through one of the other options change the impact that the feedback would have on you?

Radio-button Options:

- *Yes.*
- *No.*

If the participant indicated yes in response to the previous question**:**

How much would the impact change? Select one

Checkbox Options:

- *Minor Change in Impact.*
- *Moderate Change in Impact.*
- *Major Change in Impact.*

Question 4 – *How valuable to you is feedback from the following people?*

Question 5 – *What impact would praise from the following people have on you?*

Question 6 – *What impact would critical comments from the following people have on you?*

Response options presented to the participants in Questions 4, 5 and 6:

The participants were asked to respond to the question by reporting a score for each of six identified people on a 5-point Likert scale ranging from:

- *Not Valuable* to *Very Valuable* (question 4).
- *No Impact* to *Major Impact* (questions 5 and 6).

Identified people:

- *New Software Engineer.*
- *Experienced Software Engineer.*
- *Senior Software Engineer.*
- *Line Manager.*
- *Senior Manager.*
- *Company CEO.*

Scenario 7 – *Your line manager tells you he thought you did a good job on some recent work...*

Scenario 8 – *A fellow software engineer tells you he thought you recently wrote a really good bit of code...*

Scenario 9 – *A senior manager tells you he thought you did a good job on a recent project...*

Questions presented to the participants in Scenarios 7, 8 and 9:

Would this feedback have a positive impact on you in any of the factors listed below? Select all that apply

Checkbox Options:

- *Performance.*
- *Job Satisfaction.*
- *Motivation.*
- *Attitude.*
- *Behaviour.*
- *Feelings.*

Scenario 10 – *Your line manager tells you there is a problem and that you recently did some poor work...*

Scenario 11 – *A fellow software engineer tells you there is a problem and that you recently wrote a poor bit of code...*

Scenario 12 – *A senior manager tells you there is a problem and that you did a poor job on a recent project...*

Questions presented to the participants in Scenarios 10, 11 and 12:

Would this feedback have a negative impact on you in any of the factors listed below? Select all that apply

Checkbox Options:

- *Performance.*
- *Job Satisfaction.*
- *Motivation.*
- *Attitude.*
- *Behaviour.*
- *Feelings.*

How would you react after receiving this feedback? Select all that apply

Checkbox Options:

- *Fix the problem.*
- *Justify your actions.*
- *Discuss the problem.*
- *Do nothing.*
- *Other.*

Note: Selecting the *Other* option provided the participants with a free-text box to input additional information.

After progressing through the 12 scenarios and questions, the participants were given a free-text box to record any other comments. The participants were then asked if they would be willing to discuss their responses in a follow-up interview lasting 15-30 minutes (indicated by a *yes* or *no* selection of two radio buttons), and, if the participants indicated *yes*, then contact details and contact preferences were collected. Once the participants clicked the final button (*Complete Questionnaire*) the participant's responses were recorded in the database and a "thank-you" message was displayed. Participants who opted-in for the follow-up interview were told that they would be contacted shortly.

7.1.3 RESEARCH PREDICTIONS

Because of the basis of findings presented in earlier chapters, several predictions were formulated prior to data collection:

1. Typically, participants will report the value of feedback from a 'new software engineer' as of less value than the value of feedback from an 'experienced software engineer' or a 'senior software engineer'.

2. Positive feedback will affect job satisfaction for the majority of the respondents.
3. Negative feedback will affect feelings for the majority of the respondents.
4. Negative feedback will cause the majority of respondents to want to fix the problem.
5. If a change in medium is reported as changing the impact of the feedback, it will rarely be a major change.
6. If a participant scores feedback from software engineers as having greater value than feedback from managers in question 4, then question 5 and 6 will show a similar set of responses.
7. If a participant scores feedback from software engineers as of equal or greater value than feedback from the managers in question 4, then questions 7-12 will show a greater level of impact (more aspects selected) for the software engineer feedback.
8. If a participant scores feedback from managers as of greater value than feedback from software engineers in question 4, then questions 7-12 will show a greater level of impact (more aspects selected) for the manager feedback.

7.1.4 PILOT STUDY

The study was piloted to ensure that the questions being asked were appropriate, that the online survey operated as expected, and that the phrasing of the questions was understandable and did not cause any issues for the participants. The pilot study was completed with four participants who were all experienced software engineers working in industry. Data was collected and analysed from the pilot participants. From the pilot study, minor changes were made to the online survey. This was to ensure that the online survey was appropriate to a wider demographic of software engineers, and not specific to agile scrum-based development environments, as used in previous studies (chapters 5 and 6). As the changes were minor, the online survey was not re-piloted.

7.1.5 ETHICAL CLEARANCE

To ensure the research was appropriate and posed minimal risk to participants, ethical approval was attained from the Open University’s Human Research Ethics Committee for completing the research with human participants during this study (appendix D section 2). This was in addition to the original ethical approval that was granted for earlier studies (chapters 3, 5 and 6).

7.2 PARTICIPANTS

This study aimed to elicit responses from a larger group of participants than the previous studies (chapters 3, 5 and 6). In order to achieve this goal, several different methods of promoting the study to suitable participants were used. Flyers and business cards were produced, which were displayed and distributed at the ICSE 2012 conference. The same documents were distributed to software engineers at the annual meeting of Software Practice Advancement specialist group of the British Computer Society (BCS SPA). All the attendees at the 2012 IEEE Software Experts Summit conference also received a copy of the flyer in their conference bag.

The previously mentioned BCS SPA group also maintains a mailing list of software engineers. After extensive conversation, the chairman of the group agreed to allow an email to be sent out to the members of the mailing list introducing this research and inviting the software engineers to participant in the online survey.

The Open University has several thousand active remote students and many hundreds of students on software-related undergraduate and post-graduate courses. After seeking approval through appropriate channels and seeking permission of the leader of each module, a message was posted on several modules' online forums introducing them to this research and inviting them, if they met the identified criteria, to respond to the online survey. Members of the IT development team at the Open University also received a copy of the introduction and invitation to complete the survey.

Several participants who completed the online survey, without request or prompting, distributed the introduction to the research and an invitation to complete the survey to other friends or colleagues whom they felt would be interested and suitable for the study. This provided an unexpected but very useful avenue of contacting addition participants. It should be noted that as the survey was published in the World Wide Web, the online survey was freely accessible to anyone who knew the URL, with no password or access key required.

7.3 DATA ANALYSIS

Two forms of data were collected during this study:

- Survey data stored in the database.
- Audio data collected during the post-survey interviews.

The data collected online was stored in a database, and prior to analysis this data had to be exported and modified to be useable for analysis. The audio data had to be transcribed and associated with the participant's online survey.

7.3.1 SURVEY ANALYSIS

The first step in the survey analysis was to extract the numerical data, including the demographics of the participants and the question-by-question completion rate, defined as the number of questions for which each participant provided an answer. Following the first step, the collected data was analysed to identify emergent patterns of responses. To provide confidence in the results of the analysis and reduce the possibility of the findings being coincidental, confidence intervals were calculated and are shown on figures and graphs in section 7.4 at a 90% confidence level.

Sub-groups of participants were identified using criteria, such as the responses to the initial scenarios or questions, which were appropriate to evaluate the research predictions (section 7.1.3). For the identified groups, further analysis of remaining scenarios and questions not used during the selection criteria were used to identify emergent patterns and to evaluate the research predictions. Confidence intervals were included in the analysis of the sub-group data at the 90% confidence level to support the confidence in the identified patterns and to reduce the chance of the findings being coincidental.

7.3.2 POST-SURVEY INTERVIEW ANALYSIS

The aim of the post-survey interview was to ensure that the participants' interpretation of terms used during the online survey was as intended. While not providing direct results, this step in the analysis improved the confidence in the reliability of the findings by providing evidence that the participants interpreted each question or scenario as intended in the design of the study.

Participants related a ‘new software engineer’ to someone with little or no experience who had probably recently graduated from university, an ‘experienced software engineer’ as someone who had been developing software for multiple years, and a ‘senior software engineer’ as someone with extensive software development experience and knowledge. This interpretation was as intended by the design of the study.

7.4 RESULTS

The results of the survey provide evidence to support previous findings and further illustrate the effect of the medium and source feedback characteristics. The following sections present the results of the online survey.

7.4.1 SURVEY RESULTS

The survey was completed by 157 participants who consider themselves to be software engineers, and who reported that they work in a team environment and typically write code. The survey results are first presented as raw numerical data from all of the participants, and then are presented using selective groupings based on pre-set criteria.

Included in Figures 7.1-7.16 are visual representations (black lines) of the confidence interval range for each result. These lines indicate the expected range that the results to the same question would likely fall within if the study were to be repeated. The confidence interval ranges shown on Figures 7.1-7.7 are calculated with a 90% confidence level. The confidence intervals reduce the chance of the findings being due to coincidence. Table 7.5 includes the confidence intervals for each response, but is represented textually.

PARTICIPANT DEMOGRAPHICS

In Table 7.4 the demographic information of the participants is presented. To aid interpretation, keys to the type of response (Select One or Multiple Choice) and the response method (Checkbox Options, Dropdown Selection Box, Radio Buttons) required are included. For each of the questions, an additional column is included entitled ‘Empty’, this indicates the percentage of participants who did not indicate a response to the question. In the demographic information table, each question is represented by a short phrase. At the bottom of the demographics table, a key is provided to indicate the full question to which the participant was asked to provide a response.

TABLE 7.4 – DEMOGRAPHICS OF 157 PARTICIPANTS

Education [1]	Select One – Radio Buttons					
	GCSE	**A– Level**	**Bachelor's Degree**	**Master's Degree**	**Doctorate Degree**	**Empty**
	6%	27%	48%	14%	2%	3%

Experience [2]	Select One – Dropdown Selection Box									
	<6 Months	**6 Months**	**1– 2 years**	**3– 5 years**	**6– 10 years**	**11– 15 years**	**16– 20 years**	**21– 30 years**	**30+ years**	**Empty**
	1%	4%	8%	15%	18%	18%	13%	12%	11%	0%

At Current Job [3]	Select One – Dropdown Selection Box									
	<6 Months	**6 Months**	**1– 2 years**	**3– 5 years**	**6– 10 years**	**11– 15 years**	**16– 20 years**	**21– 30 years**	**30+ years**	**Empty**
	8%	6%	25%	24%	13%	12%	2%	3%	2%	5%

Team Size [4]	Select One – Dropdown Selection Box							
	Solo	**2– 3 people**	**4– 6 people**	**7– 9 people**	**10– 15 people**	**16– 30 people**	**30+ people**	**Empty**
	10%	17%	29%	11%	17%	8%	8%	1%

Other Engineers [5]	Select One – Dropdown Selection Box								
	Just me	**1– 2 other SEs**	**3– 5 other SEs**	**6– 10 other SEs**	**11– 15 other SEs**	**16– 20 other SEs**	**21– 30 other SEs**	**30+ other SEs**	**Empty**
	16%	28%	26%	19%	1%	6%	0%	6%	1%

Development Methods [6]	Multiple Choice – Checkbox Options					
	Agile Methods	**Waterfall Model**	**Spiral Model**	**Iterative Development**	**Other Methods**	**Empty**
	64%	71%	10%	74.52%	13%	2%

Typical Tasks [7]	Multiple Choice – Checkbox Options						
	Requirements Gathering	**Systems Design**	**Writing Code**	**Systems Testing**	**Systems Maintenance**	**Other Tasks**	**Empty**
	61%	71%	92%	68%	71%	10%	1%

Table 7.4 Question Key:

1 – What is the highest level you are currently educated to?

2 – Approximately, how many years' experience do you have developing software?

3 – Approximately, how long have you been at your current employer?

4 – Including software engineers and other colleagues, what size development team are you typically in?

5 – How many other software engineers are typically in your development team?

6 – Which of the following types of development methods have you had experience with?

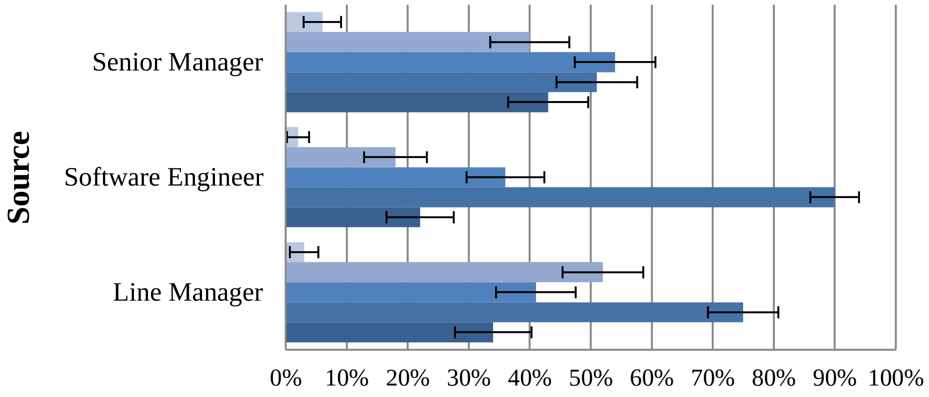
7 – Which of the following tasks are typical of your current job?

NUMERICAL RESULTS

Of the 157 total participants, 144 provided a response to all of the first 6 scenarios and questions, indicating a participant completion rate of 92%. Due to the nature of scenarios 7-12, a lack of a response does not indicate that the participant did not complete the scenario; it could simply mean that there was no impact on the listed aspects and subsequently an impact would not need to be indicated. 114 participants indicated a response across all 12 scenarios and questions, indicating that 73% of the participants provided a response to every scenario and question.

The first three scenarios in the survey presented the participants with feedback from their line manager [1] (*Your line manager tells you he thought you did a good job on some recent work...*), a software engineer [2] (*A fellow software engineer tells you he thought you recently wrote a really good bit of code...*), and a senior manager [3] (*A senior manager tells you he thought you did a good job on a recent project...*). The participants were asked to indicate through which of 5 listed media they would prefer to receive the feedback. Figure 7.1 presents the percentage of participants that indicated a preference for each medium for each of the three questions, showing that the participants reported difference preferences for the medium used to receive feedback depending on the source of the feedback. Please note that participants were able to select multiple media in response to each scenario.

In Figure 7.1, and all subsequent figures in this section with confidence intervals, the presented data shows the percentage of respondents who indicated a specific answer to each question. Confidence intervals, as shown with black lines on each figure, indicate the range within which we would expect the responses to fall if this study were to be repeated. In this research, the results can be said to be robust, and not due to coincidence, where the confidence intervals do not overlap between different answers to one question. For example, in Figure 7.1 we can be confident that the participants' preference to receiving feedback through casual chat is different dependent upon the source of the feedback, as there is no overlap in the confidence interval range for all three different sources (75%, 90%, and 51% respectively).



Medium	Line Manager	Software Engineer	Senior Manager
■ Notice Board	3%	2%	6%
■ One-to-one Meeting	52%	18%	40%
■ Email	41%	36%	54%
■ Casual Chat	75%	90%	51%
■ Team Meeting	34%	22%	43%

FIGURE 7.1 – EXPRESSED PREFERENCE FOR MEDIUM FROM DIFFERENT SOURCE (QUESTIONS 1-3)

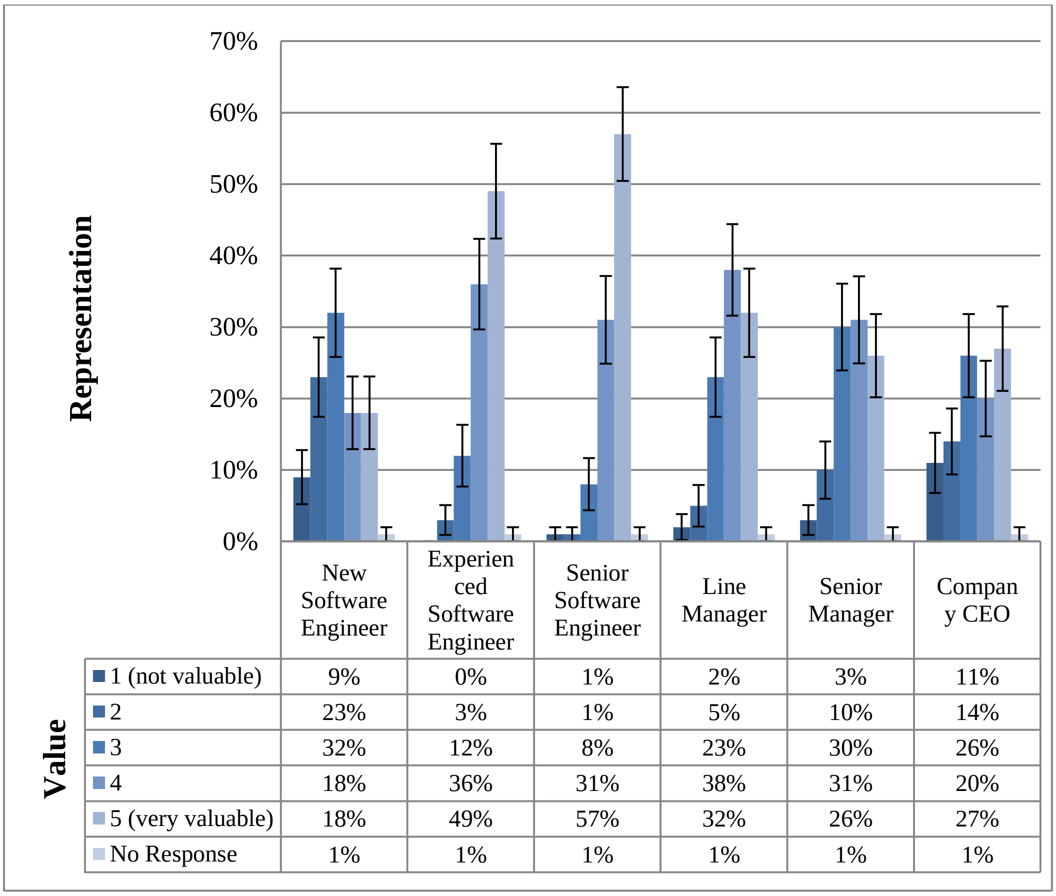
In addition to indicating a medium preference for scenarios 1-3, the participants were asked to indicate if a change in the medium used to provide the feedback would alter the impact of the feedback [4] (*Would receiving this feedback through one of the other options change the impact that the feedback would have on you?*).

TABLE 7.5 – ENGINEERS REPORTING THAT A DIFFERENT MEDIUM WOULD CHANGE THE IMPACT OF RECEIVED FEEDBACK, AND HOW MUCH

	Yes	Select One (from yes's)		
Feedback	**Impact change** [4]	**Minor** [5]	**Moderate** [5]	**Major** [5]
Line Manager [1]	52% (45.61%-58.84%)	38% (28.84%-46.76%)	46% (37.12%-55.56%)	16% (9.10%-22.60%)
Software Engineer [2]	42% (35.49%-48.57%)	44% (33.66%-54.21%)	47% (36.66%-57.29%)	9% (3.14%-15.04%)
Senior Manager [3]	53% (46.25%-59.47%)	33% (23.92%-41.13%)	48% (39.01%-57.37%)	19% (12.02%-26.52%)[4]

[4] The table includes the confidence intervals shown inside brackets below each response.

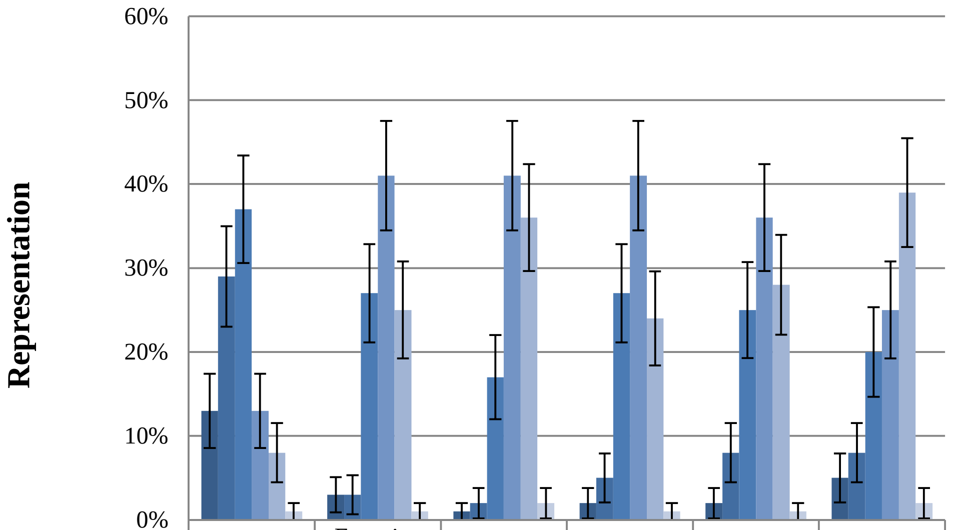
Table 7.5 shows the percentage of participants who indicated that a change in the medium would alter the impact of the feedback, including the confidence interval of the responses (90% confidence). Table 7.5 also displays the response to the follow-on question assessing the level of change [5] (*How much would the impact change?*) of all participants who indicated "yes" to the initial question. Table 7.5 shows that half of the participants indicated that a change in the medium of received feedback would change the impact, but less than 20% of those indicating a change in impact (10% of the total participants) reported that they believed the change in impact would be major.



	New Software Engineer	Experienced Software Engineer	Senior Software Engineer	Line Manager	Senior Manager	Company CEO
■ 1 (not valuable)	9%	0%	1%	2%	3%	11%
■ 2	23%	3%	1%	5%	10%	14%
■ 3	32%	12%	8%	23%	30%	26%
■ 4	18%	36%	31%	38%	31%	20%
■ 5 (very valuable)	18%	49%	57%	32%	26%	27%
■ No Response	1%	1%	1%	1%	1%	1%

FIGURE 7.2 – REPORTED VALUE OF FEEDBACK FROM DIFFERENT SOURCE (QUESTION 4)

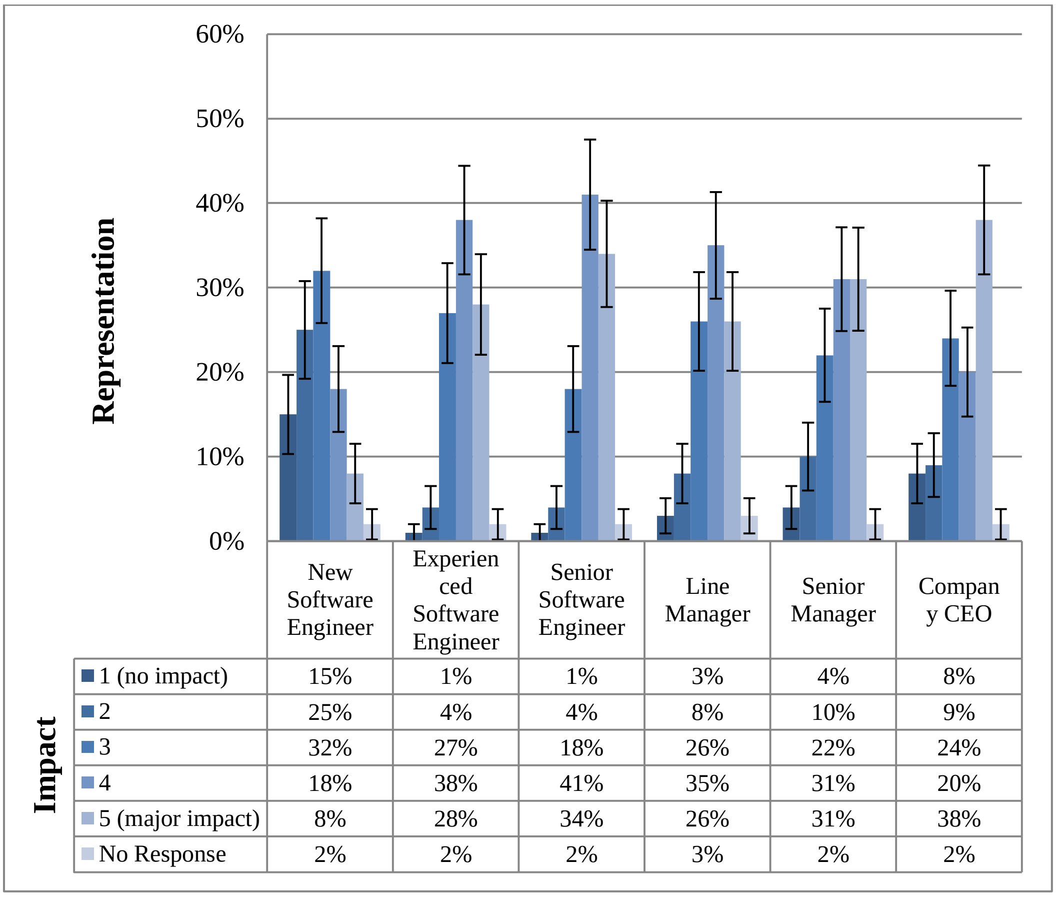
Figure 7.2 shows the responses to question 4, which presented the participants with short descriptions of six different people, and asked them to rate how valuable their feedback was to the participant (*How valuable to you is feedback from the following people?*) on a Likert scale ranging from 1 (not valuable) to 5 (valuable). Figure 7.2 displays that the participants reported a variation in the value of feedback depending on the source of the feedback.



Impact	New Software Engineer	Experienced Software Engineer	Senior Software Engineer	Line Manager	Senior Manager	Company CEO
■ 1 (no impact)	13%	3%	1%	2%	2%	5%
■ 2	29%	3%	2%	5%	8%	8%
■ 3	37%	27%	17%	27%	25%	20%
■ 4	13%	41%	41%	41%	36%	25%
■ 5 (major impact)	8%	25%	36%	24%	28%	39%
■ No Response	1%	1%	2%	1%	1%	2%

FIGURE 7.3 – REPORTED IMPACT OF PRAISE FROM DIFFERENT SOURCE (QUESTION 5)

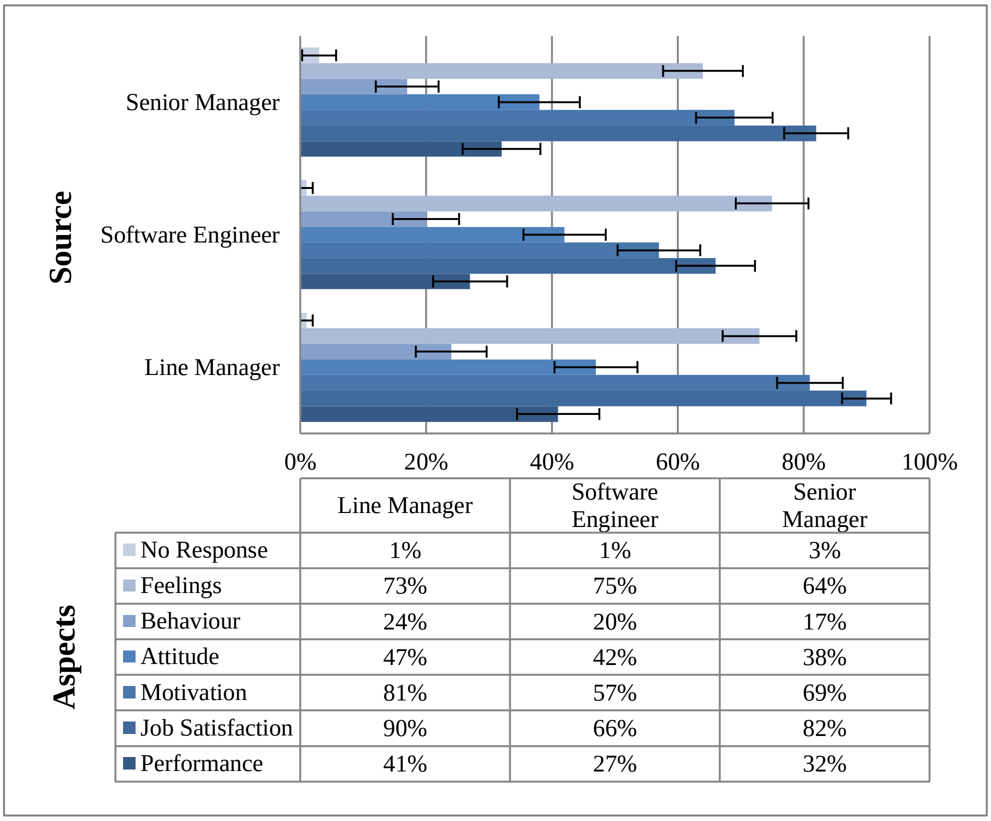
Figure 7.3 shows the results of the participants' responses to question 5. Using the same short descriptions of six people as were presented to the participants in question 4, question 5 asked the participants to rate the impact of praise on the participant (*What impact would praise from the following people have on you?*) on a Likert scale ranging from 1 (no impact) to 5 (major impact). Figure 7.3 displays that the participants reported a variation in the impact of received praise depending on the source of the praise.



Impact	New Software Engineer	Experienced Software Engineer	Senior Software Engineer	Line Manager	Senior Manager	Company CEO
■ 1 (no impact)	15%	1%	1%	3%	4%	8%
■ 2	25%	4%	4%	8%	10%	9%
■ 3	32%	27%	18%	26%	22%	24%
■ 4	18%	38%	41%	35%	31%	20%
■ 5 (major impact)	8%	28%	34%	26%	31%	38%
■ No Response	2%	2%	2%	3%	2%	2%

FIGURE 7.4 – REPORTED IMPACT OF CRITICAL COMMENTS FROM DIFFERENT SOURCE (QUESTION 6)

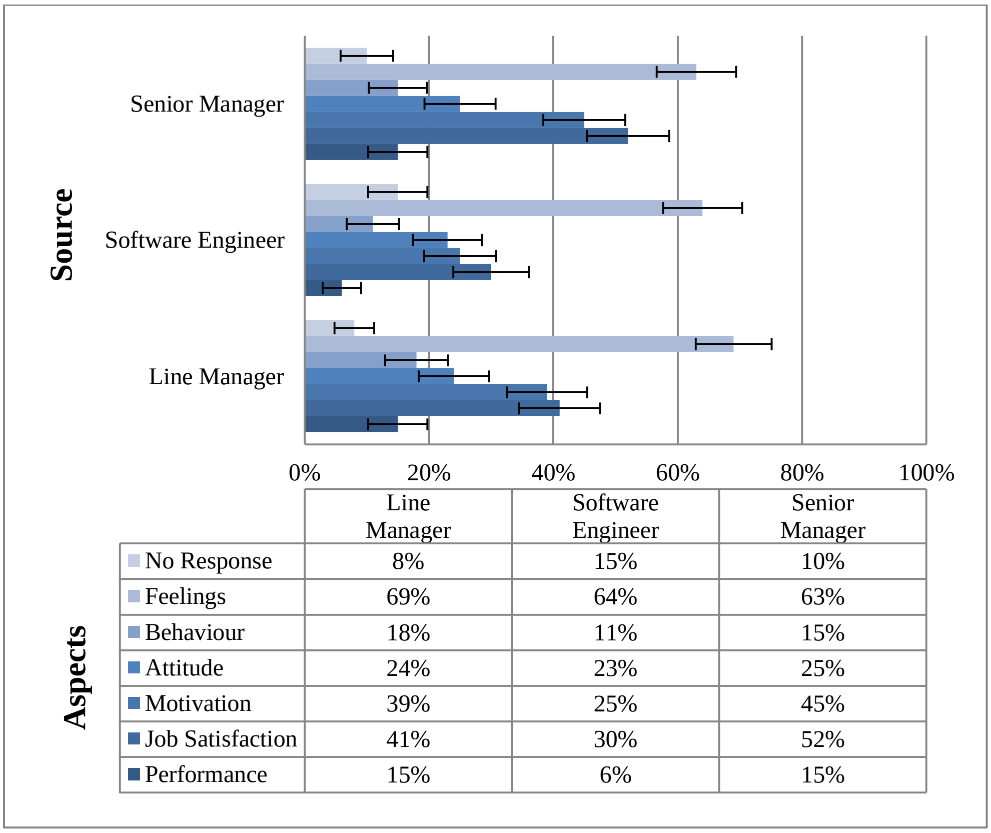
Figure 7.4 shows the results of the participants’ responses to question 6. Using the same short descriptions of six people as were presented to the participants in question 4 and 5, question 6 asked the participants to rate the impact of critical comments on the participant (*What impact would critical comments from the following people have on you?*) on a Likert scale ranging from 1 (no impact) to 5 (major impact). Figure 7.4 displays that the participants reported a variation in the impact of received critical comments depending on the source of the critical comments.



	Line Manager	Software Engineer	Senior Manager
■ No Response	1%	1%	3%
■ Feelings	73%	75%	64%
■ Behaviour	24%	20%	17%
■ Attitude	47%	42%	38%
■ Motivation	81%	57%	69%
■ Job Satisfaction	90%	66%	82%
■ Performance	41%	27%	32%

FIGURE 7.5 – ASPECTS POSITIVELY IMPACTED BY FEEDBACK RECEIVED FROM DIFFERENT SOURCE (QUESTIONS 7-9)

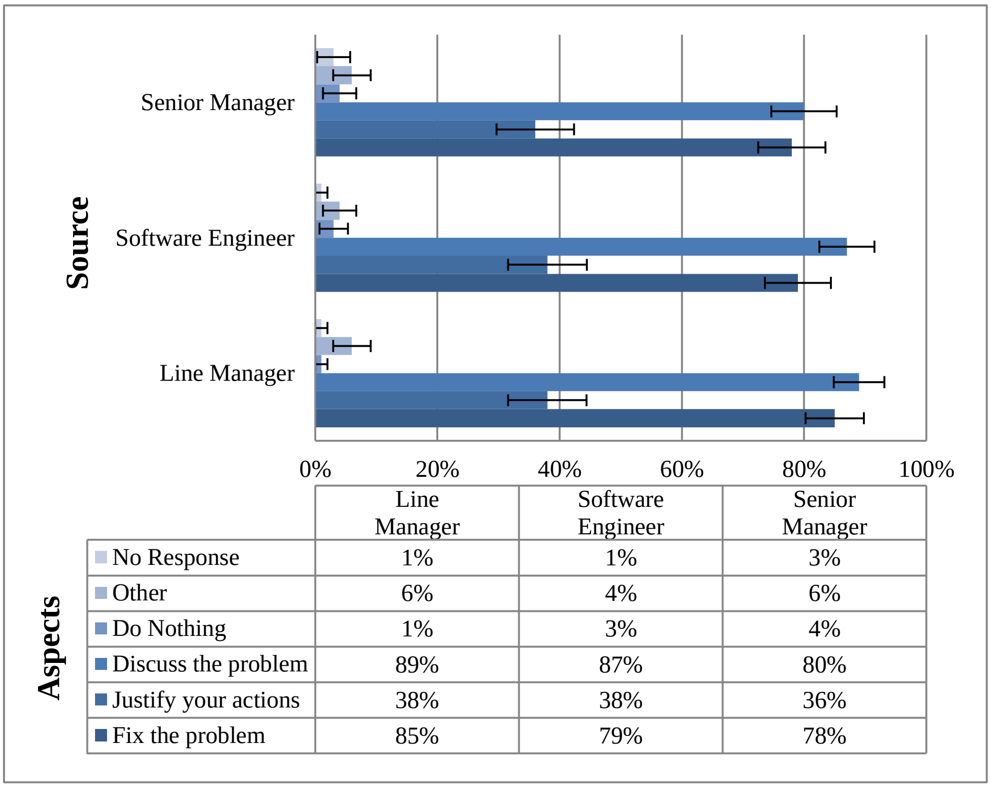
Using the same feedback examples as were presented in scenarios 1-3, scenarios 7-9 presented the participants with feedback from their line manager [1] (*Your line manager tells you he thought you did a good job on some recent work...*), a software engineer [2] (*A fellow software engineer tells you he thought you recently wrote a really good bit of code...*), and a senior manager [3] (*A senior manager tells you he thought you did a good job on a recent project...*). The participants were asked to indicate if receiving this feedback would have a positive impact on them in any of six aspects (*Performance, Job Satisfaction, Motivation, Attitude, Behaviour and Feelings*). Figure 7.5 presents the percentage of participants who indicated a positive impact for each aspect when receiving each different instance of feedback, including a column indicating the percentage of participants who did not indicate an impact on any of the six listed aspects. Figure 7.5 displays that the aspects reported as being affected by the participants varied for some of the aspects depending on the source of the positive feedback. Please note that participants were able to select multiple impacted aspects in response to each scenario.



	Line Manager	Software Engineer	Senior Manager
No Response	8%	15%	10%
Feelings	69%	64%	63%
Behaviour	18%	11%	15%
Attitude	24%	23%	25%
Motivation	39%	25%	45%
Job Satisfaction	41%	30%	52%
Performance	15%	6%	15%

FIGURE 7.6 – ASPECTS NEGATIVELY IMPACTED FROM FEEDBACK RECEIVED FROM DIFFERENT SOURCE (QUESTIONS 10-12)

Similar to scenarios 1-3 and 7-9, scenarios 10-12 presented the participants with feedback from their line manager [6] (*Your line manager tells you there is a problem and that you recently did some poor work...*), a software engineer [7] (*A fellow software engineer tells you there is a problem and that you recently wrote a poor bit of code...*), and a senior manager [8] (*A senior manager tells you there is a problem and that you did a poor job on a recent project...*). The participants were asked to indicate if receiving this feedback would have a negative impact on them in any of six aspects (*Performance, Job Satisfaction, Motivation, Attitude, Behaviour and Feelings*). Figure 7.6 presents the percentage of participants who indicated a negative impact on each aspect when receiving each different instance of feedback, including a column indicating the percentage of participants who did not indicate an impact on any of the six aspects. Figure 7.6 displays that the aspects reported as being affected by the participants varied for some of the aspects depending on the source of the negative feedback Please note participants were able to select multiple aspects in response to each scenario.



	Line Manager	Software Engineer	Senior Manager
■ No Response	1%	1%	3%
■ Other	6%	4%	6%
■ Do Nothing	1%	3%	4%
■ Discuss the problem	89%	87%	80%
■ Justify your actions	38%	38%	36%
■ Fix the problem	85%	79%	78%

FIGURE 7.7 – REPORTED REACTION TO NEGATIVE FEEDBACK RECEIVED FROM DIFFERENT SOURCE (QUESTION 10-12)

In addition to indicating if the feedback would have a negative impact on any of the six aspects, the participants were asked to report how they would react after receiving this feedback (*How would you react after receiving this feedback?*). In Figure 7.7, the percentage of participants that indicated a reaction to receiving the critical comments is shown, including a column indicating the percentage of participants who did not indicate an impact on any of the 5 listed reactions. Figure 7.7 displays that the participants' reactions to receiving negative feedback did not significantly vary depending on the source of the feedback. Please note that participants were able to select multiple reactions in response to each question.

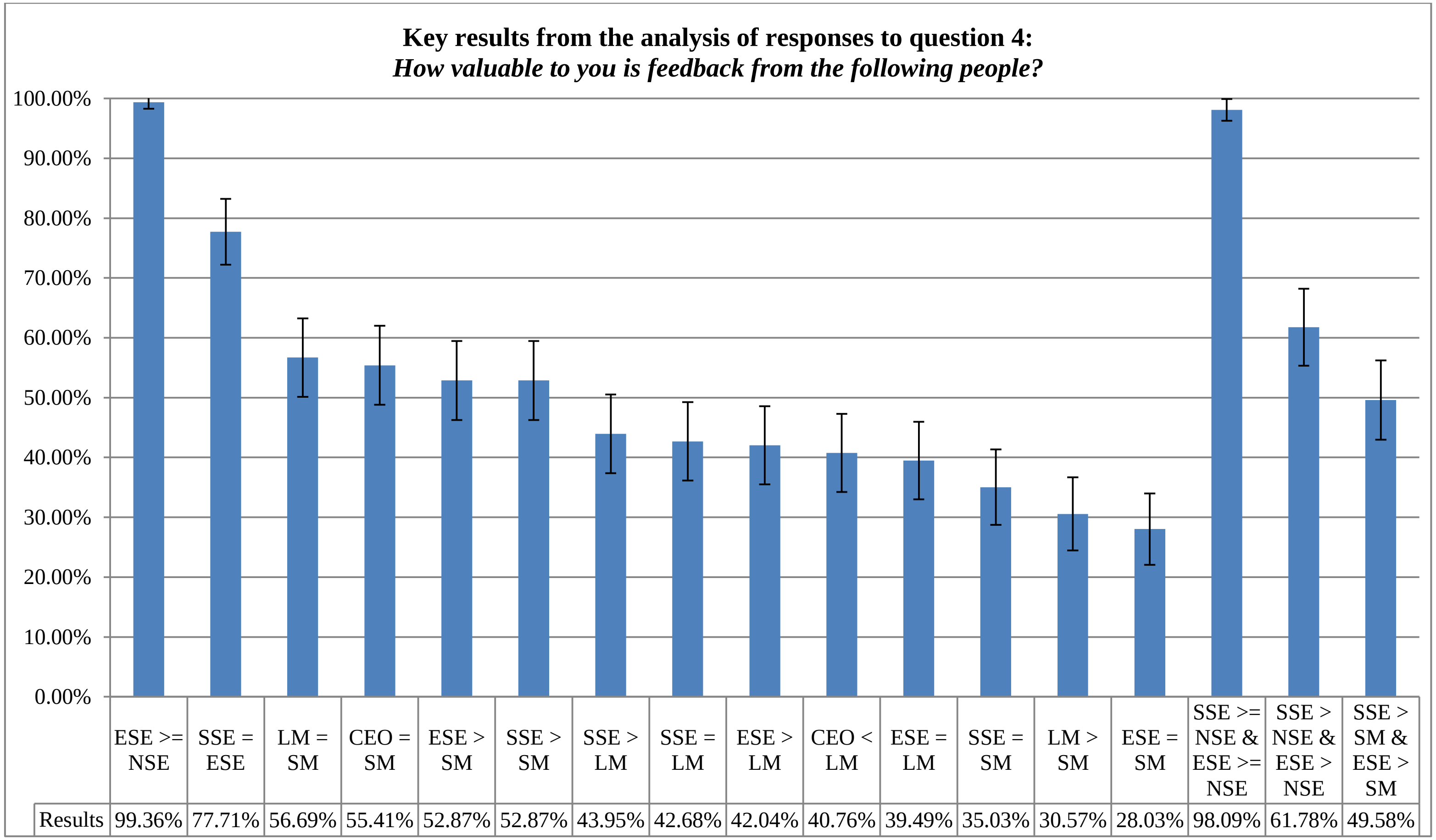
COMPARATIVE ANALYSIS

After extracting the initial numerical results, further analysis of the collected data focused on comparing the data to the pre-collection predictions (section 7.1.3). The following analysis investigates data relevant to the data predictions, while also investigating any emerging themes and patterns.

Due to the number of different combinations of responses able to be analysed, the analysis of the data is vast and contains a wide range of queries. In total, while comparing responses to questions 4, 5 and 6 using all possible combinations of the

6 sources and all comparison combinations (greater than, equal to, less than, greater than or equal to, less than or equal to) there were a total of 110,936 different queries performed on the data, with 10,896 of the queries found as representative of 10% or more of the participants. A concise summary of these results, listing the main patterns identified from the comparative analysis, is presented in Figure 7.8, Figure 7.9, Figure 7.10, and Table 7.9. A more comprehensive listing can be found in appendix D section 3. Not all 10,896 results are included in the appendix as they represent duplicate or very similar results.

The following figures present the key results from the analysis of questions 4, 5 and 6. A key is provided on the page after each figure.



	ESE >= NSE	SSE = ESE	LM = SM	CEO = SM	ESE > SM	SSE > SM	SSE > LM	SSE = LM	ESE > LM	CEO < LM	ESE = LM	SSE = SM	LM > SM	ESE = SM	SSE >= NSE & ESE >= NSE	SSE > NSE & ESE > NSE	SSE > SM & ESE > SM
Results	99.36%	77.71%	56.69%	55.41%	52.87%	52.87%	43.95%	42.68%	42.04%	40.76%	39.49%	35.03%	30.57%	28.03%	98.09%	61.78%	49.58%

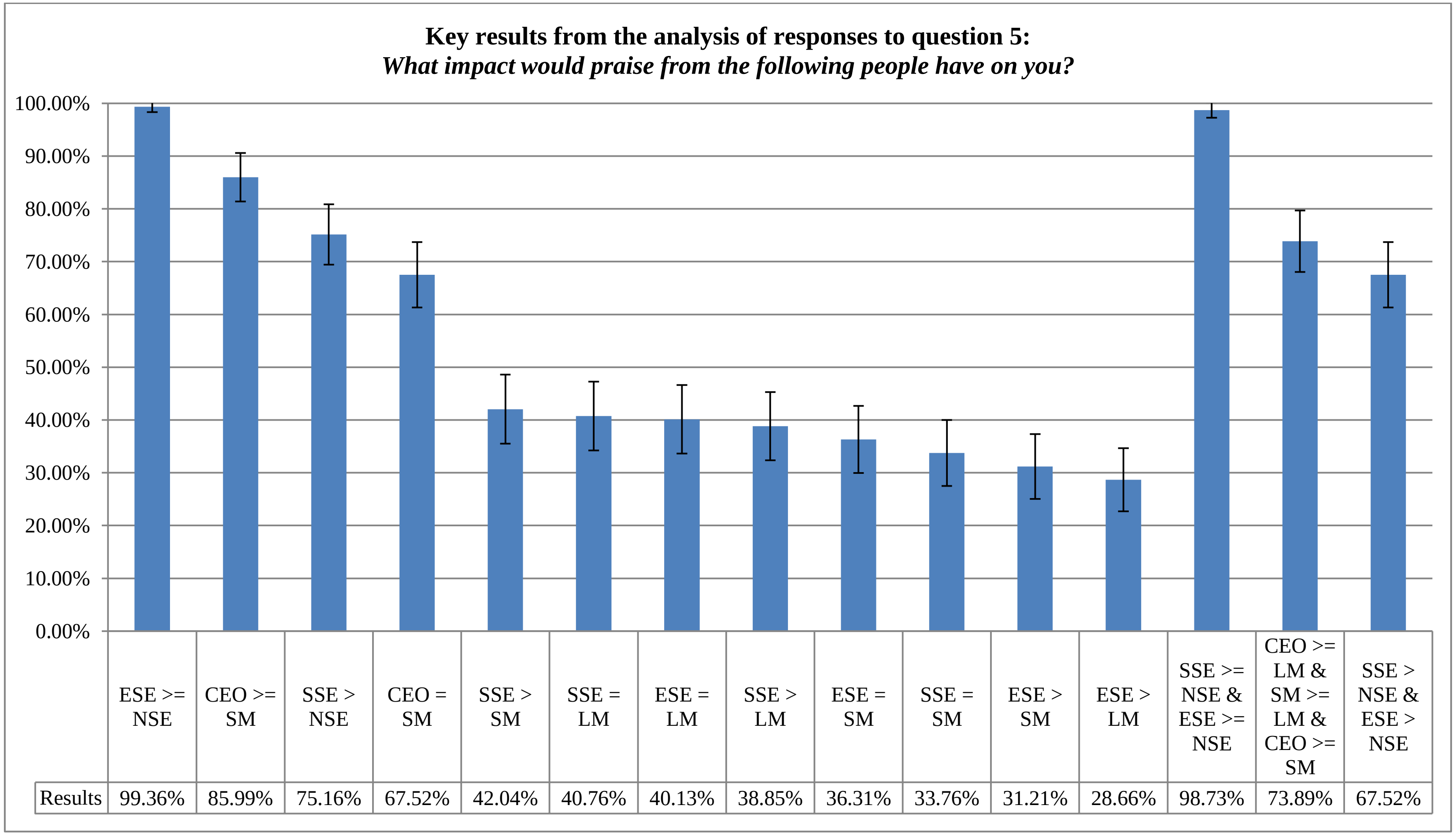
FIGURE 7.8 – KEY RESULTS FROM THE ANALYSIS OF RESPONSES TO QUESTION FOUR

Acronym Key:

NSE: New Software Engineer	**ESE**: Experienced Software Engineer	**SSE**: Senior Software Engineering
LM: Line Manager	**SM**: Senior Manager	**CEO**: Company CEO

TABLE 7.6 – FIGURE 7.15 FULL RESULTS KEY

Key	Statement
ESE >= NSE	feedback from an experienced software engineer *is of equal or greater value to me than* feedback from a new software engineer
SSE = ESE	feedback from a senior software engineer *is of equal value to me as* feedback from an experienced software engineer
LM = SM	feedback from a line manafer *is of equal value to me as* feedback from a senior manager
CEO = SM	feedback from the company ceo *is of equal value to me as* feedback from a senior manager
ESE > SM	feedback from an experienced software engineer *is of greater value to me as* feedback from a senior manager
SSE > SM	feedback from a senior software engineer *is of greater value to me as* feedback from a senior manager
SSE > LM	feedback from a senior software engineer *is of greater value to me as* feedback from a line manager
SSE = LM	feedback from a senior software engineer *is of equal value to me as* feedback from a line manager
ESE > LM	feedback from an experienced software engineer *is of greater value to me as* feedback from a line manager
CEO < LM	feedback from the company ceo *is of less value to me than* feedback from a line manager
ESE = LM	feedback from a experienced software engineer *is of equal value to me as* feedback from an line manager
SSE = SM	feedback from a senior software engineer *is of equal value to me as* feedback from a senior manager
LM > SM	feedback from a line manafer *is of greater value to me than* feedback from a senior manager
ESE = SM	feedback from an experienced software engineer *is of equal value to me as* feedback from a senior manager
SSE >= NSE & ESE >= NSE	feedback from a senior software engineer *is of equal or greater value to me than* feedback from a new software engineer **AND** feedback from an experienced software engineer *is of equal or greater value to me than* feedback from a new software engineer
SSE > NSE & ESE > NSE	feedback from a senior software engineer *is of greater value to me than* feedback from a new software engineer **AND** feedback from an experienced software engineer *is of greater value to me than* feedback from a new software engineer
SSE > SM & ESE > SM	feedback from a senior software engineer *is of greater value to me as* feedback from a senior manager **AND** feedback from an experienced software engineer *is of greater value to me as* feedback from a senior manager



	ESE >= NSE	CEO >= SM	SSE > NSE	CEO = SM	SSE > SM	SSE = LM	ESE = LM	SSE > LM	ESE = SM	SSE = SM	ESE > SM	ESE > LM	SSE >= NSE & ESE >= NSE	CEO >= LM & SM >= LM & CEO >= SM	SSE > NSE & ESE > NSE
Results	99.36%	85.99%	75.16%	67.52%	42.04%	40.76%	40.13%	38.85%	36.31%	33.76%	31.21%	28.66%	98.73%	73.89%	67.52%

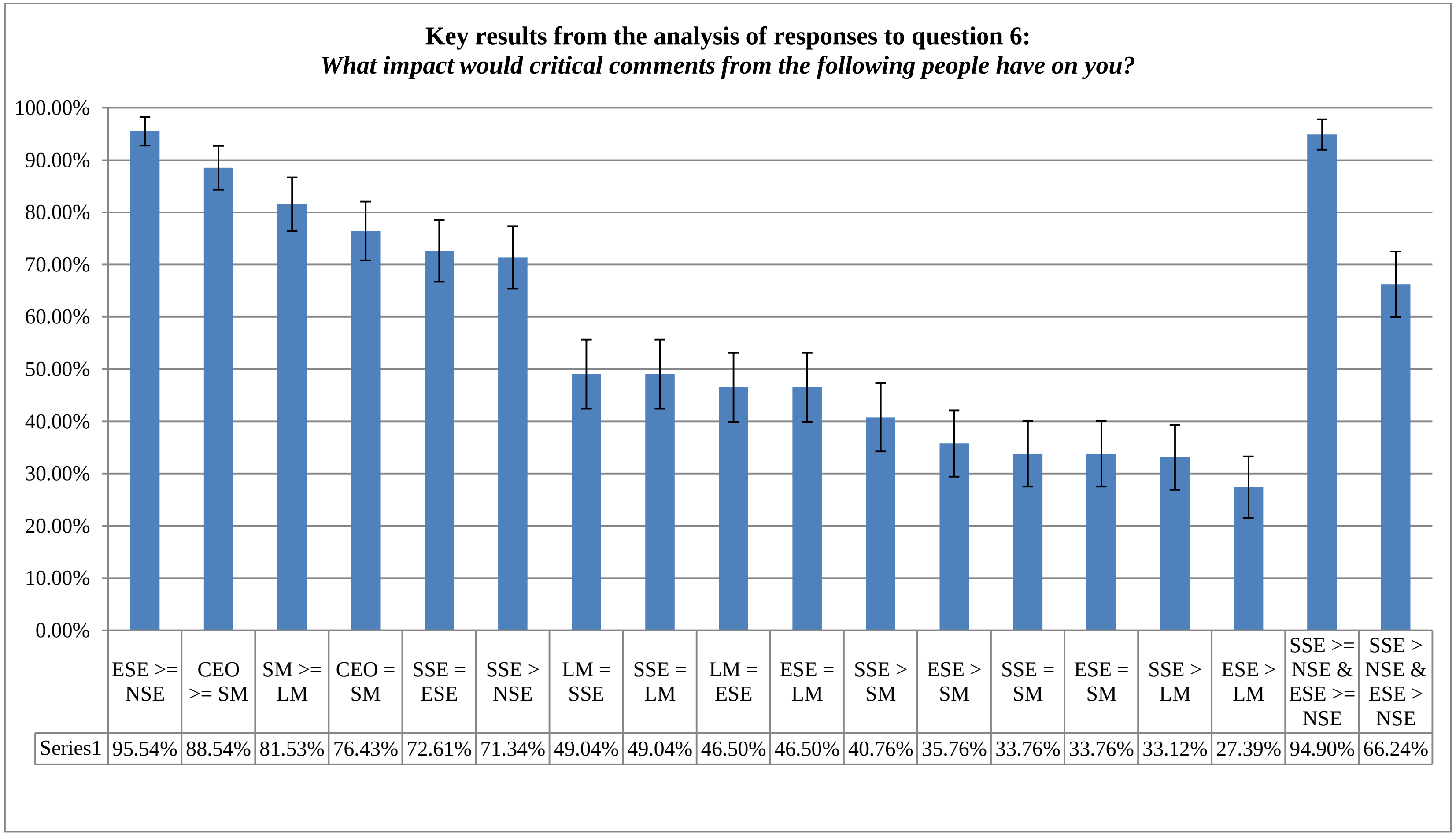
FIGURE 7.9 – KEY RESULTS FROM THE ANALYSIS OF RESPONSES TO QUESTION FIVE

Acronym Key:

NSE: New Software Engineer
ESE: Experienced Software Engineer
SSE: Senior Software Engineering
LM: Line Manager
SM: Senior Manager
CEO: Company CEO

TABLE 7.7 – FIGURE 7.16 FULL RESULTS KEY

Key	Statement
SSE > NSE	praise from a senior software engineer *has greater impact on me than* praise from a new software engineer
SSE > NSE & ESE > NSE	praise from a senior software engineer *has greater impact on me than* praise from a new software engineer **AND** praise from an experienced software engineer *has greater impact on me than* praise from a new software engineer
SSE >= NSE & ESE >= NSE	praise from a senior software engineer *has equal or greater impact on me than* praise from a new software engineer **AND** praise from an experienced software engineer *has equal or greater impact on me than* praise from a new software engineer
ESE >= NSE	praise from an experienced software engineer *has equal or greater impact on me than* praise from a new software engineer
CEO >= SM	praise from the company ceo *has equal or greater impact on me than* praise from a senior manager
CEO >= LM & SM >= LM & CEO >= SM	praise from the company ceo *has equal or greater impact on me than* praise from a line manager **AND** praise from a senior manager *has equal or greater impact on me than* praise from a line manager **AND** praise from the company ceo *has equal or greater impact on me than* praise from a senior manager
CEO = SM	praise from the company ceo *has equal impact on me as* praise from a senior manager
SSE > SM	praise from a senior software engineer *has greater impact on me as* praise from a senior manager
SSE = SM	praise from a senior software engineer *has equal impact on me as* praise from a senior manager
SSE > LM	praise from a senior software engineer *has greater impact on me as* praise from a line manager
SSE = LM	praise from a senior software engineer *has equal impact on me as* praise from a line manager
ESE > SM	praise from an experienced software engineer *has greater impact on me as* praise from a senior manager
ESE = SM	praise from an experienced software engineer *has equal impact on me as* praise from a senior manager
ESE > LM	praise from an experienced software engineer *has greater impact on me as* praise from a line manager
ESE = LM	praise from an experienced software engineer *has equal impact on me as* praise from a line manager



Key results from the analysis of responses to question 6:
What impact would critical comments from the following people have on you?

	ESE >= NSE	CEO >= SM	SM >= LM	CEO = SM	SSE = ESE	SSE > NSE	LM = SSE	SSE = LM	LM = ESE	ESE = LM	SSE > SM	ESE > SM	SSE = SM	ESE = SM	SSE > LM	ESE > LM	SSE >= NSE & ESE >= NSE	SSE > NSE & ESE > NSE
Series1	95.54%	88.54%	81.53%	76.43%	72.61%	71.34%	49.04%	49.04%	46.50%	46.50%	40.76%	35.76%	33.76%	33.76%	33.12%	27.39%	94.90%	66.24%

FIGURE 7.10 – KEY RESULTS FROM THE ANALYSIS OF PARTICIPANT RESPONSES TO QUESTION SIX

Acronym Key:

NSE: New Software Engineer	**ESE**: Experienced Software Engineer	**SSE**: Senior Software Engineering
LM: Line Manager	**SM**: Senior Manager	**CEO**: Company CEO

TABLE 7.8 – FIGURE 7.17 FULL RESULTS KEY

Key	Statement
ESE >= NSE	critical comments from an experienced software engineer *have equal or greater impact on me than* critical comments from a new software engineer
CEO >= SM	critical comments from the company ceo *have equal or greater impact on me than* critical comments from a senior manager
SM >= LM	critical comments from a senior manager *have equal or greater impact on me than* critical comments from a line manager
CEO = SM	critical comments from the company ceo *have equal impact on me as* critical comments from a senior manager
SSE = ESE	critical comments from a senior software engineer *has equal impact on me as* critical comments from an experienced software engineer
SSE > NSE	critical comments from a senior software engineer *have greater impact on me than* critical comments from a new software engineer
LM = SSE	critical comments from a line manager *has equal impact on me as* critical comments from a senior software engineer
SSE = LM	critical comments from a senior software engineer *has equal impact on me as* critical comments from a line manager
LM = ESE	critical comments from a line manager *has equal impact on me as* critical comments from an experienced software engineer
ESE = LM	critical comments from an experienced software engineer *has equal impact on me as* critical comments from a line manager
SSE > SM	critical comments from a senior software engineer *has greater impact on me as* critical comments from a senior manager
ESE > SM	critical comments from an experienced software engineer *has greater impact on me as* critical comments from a senior manager
SSE = SM	critical comments from a senior software engineer *has equal impact on me as* critical comments from a senior manager
ESE = SM	critical comments from an experienced software engineer *has equal impact on me as* critical comments from a senior manager
SSE > LM	critical comments from a senior software engineer *has greater impact on me as* critical comments from a line manager
ESE > LM	critical comments from an experienced software engineer *has greater impact on me as* critical comments from a line manager
SSE >= NSE & ESE >= NSE	critical comments from a senior software engineer *have equal or greater impact on me than* critical comments from a new software engineer **AND** critical comments from an experienced software engineer *have equal or greater impact on me than* critical comments from a new software engineer
SSE > NSE & ESE > NSE	critical comments from a senior software engineer *have greater impact on me than* critical comments from a new software engineer **AND** critical comments from an experienced software engineer *have greater impact on me than* critical comments from a new software engineer

The following table presents the findings that show patterns across more than one question.

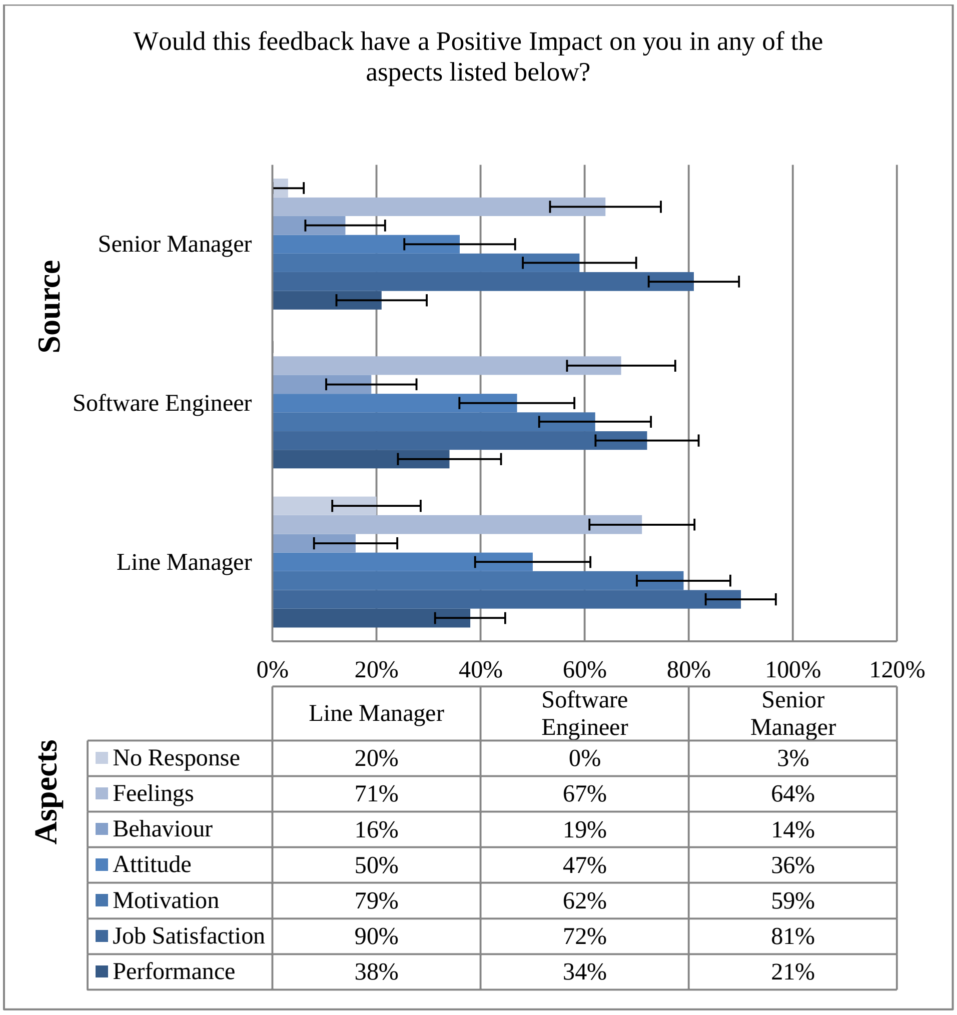
TABLE 7.9 – KEY NUMERICAL RESULTS FROM RESPONSES ACROSS COMBINATIONS OF QUESTIONS FOUR TO SIX

Feedback Value (Q4)	Praise Impact (Q5)	Critical Comments Impact (Q6)	Participants
Results investigating ‘greater than’ relationships			
	praise from a senior software engineer *has greater impact on me than* praise from a new software engineer	critical comments from a senior software engineer *have greater impact on me than* critical comments from a new software engineer	62% (98/157)
feedback from a senior software engineer *is of greater value to me than* feedback from a new software engineer	praise from a senior software engineer *has greater impact on me than* praise from a new software engineer		58% (91/157)
Results investigating ‘greater than or equal to’ relationships			
feedback from a senior software engineer *is of equal or greater value to me than* feedback from a new software engineer **AND** feedback from an experienced software engineer *is of equal or greater value to me than* feedback from a new software engineer	praise from a senior software engineer *has equal or greater impact on me than* praise from a new software engineer **AND** praise from an experienced software engineer *has equal or greater impact on me than* praise from a new software engineer		97% (153/157)
feedback from an experienced software engineer *is of equal or greater value to me than* feedback from a new software engineer	praise from an experienced software engineer *has equal or greater impact on me than* praise from a new software engineer	critical comments from an experienced software engineer *have equal or greater impact on me than* critical comments from a new software engineer	95% (149/157)
	praise from the company ceo *has equal or greater impact on me than* praise from a senior manager	critical comments from the company ceo *have equal or greater impact on me than* critical comments from a senior manager	80% (126/157)
Results investigating ‘equal to’ relationships			
feedback from a senior software engineer *is of equal value to me as* feedback from an experienced software engineer		critical comments from a senior software engineer *has equal impact on me as* critical comments from an experienced software engineer	62% (97/157)
feedback from the company ceo *is of equal value to me as* feedback from a senior manager		critical comments from the company ceo *has equal impact on me as* critical comments from a senior manager	48% (76/157)

The results show that 60% of the participants reported feedback from a senior software engineer as more valuable than feedback from a new software engineer, and feedback from an experienced software engineer more valuable than feedback from a new software engineer. This suggests that the value and impact of feedback received from an experienced or senior software engineer is of a similar value, and that feedback received from either a senior or experienced software engineer will have more impact than feedback received from a new software engineer. It should be noted that only one participant reported that the impact and value of feedback would go down, as the experience of the feedback source increased (new-experienced-senior).

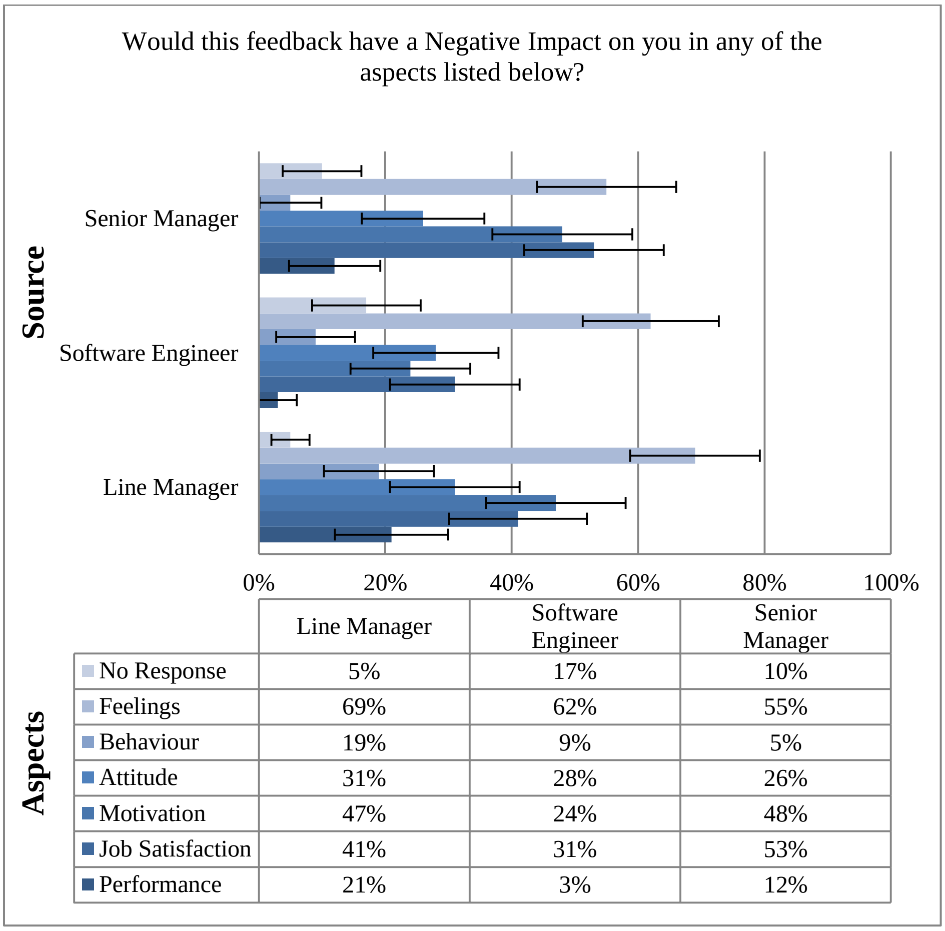
The analysis investigated the relationship between the participant’s reported feedback value, and the impact feedback from different sources may have on the participant. This was done by investigating sub-sets of participants identified from their responses to question 4 and evaluating their responses to future questions, specifically questions 7 to 12.

Investigation 1: Participants who reported a higher value for feedback from an experienced or senior software engineer compared to both a line manager and senior manager. 37% (58/157) of the participants were represented by this sub-group. The results of the impact reported by the participants in response to questions 7 to 12 is shown in Figure 7.11 and Figure 7.12. These findings are discussed and used in section 7.6.1 to support or dispute the research predictions presented in section 7.1.3.



	Line Manager	Software Engineer	Senior Manager
No Response	20%	0%	3%
Feelings	71%	67%	64%
Behaviour	16%	19%	14%
Attitude	50%	47%	36%
Motivation	79%	62%	59%
Job Satisfaction	90%	72%	81%
Performance	38%	34%	21%

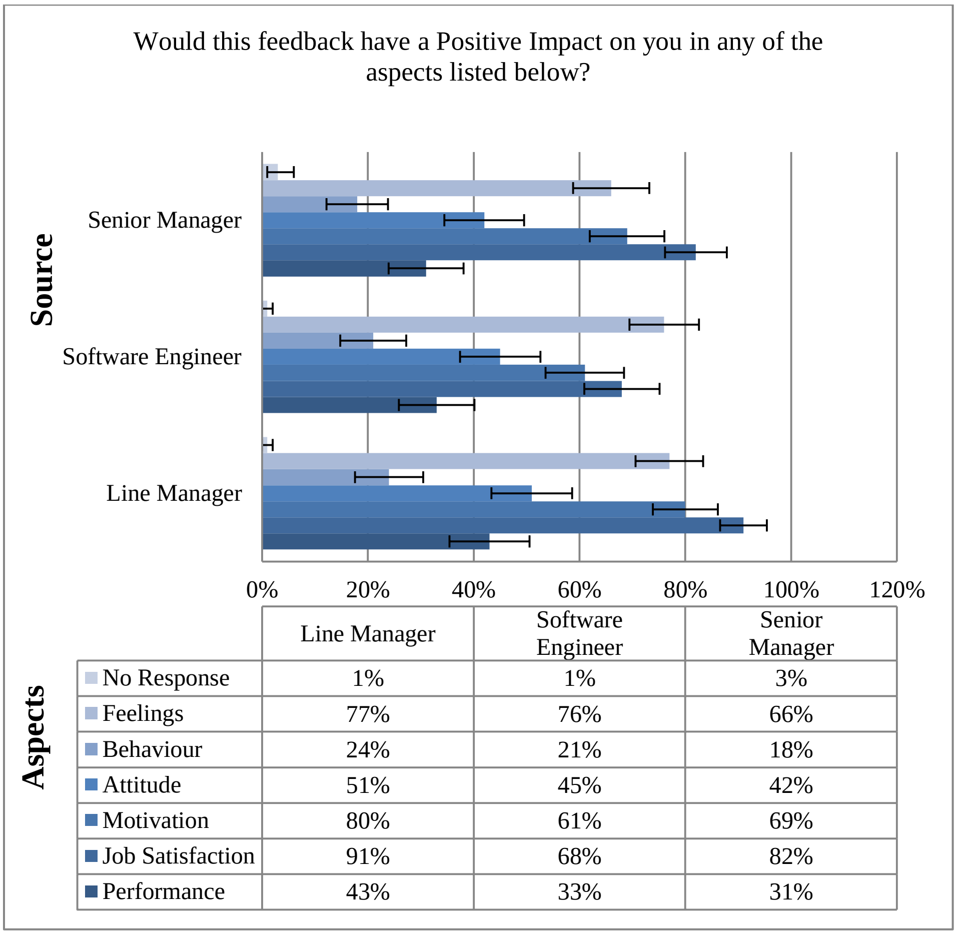
FIGURE 7.11 – INVESTIGATION 1, QUESTIONS 7-9: SUB-GROUP REPRESENTATION



	Line Manager	Software Engineer	Senior Manager
■ No Response	5%	17%	10%
■ Feelings	69%	62%	55%
■ Behaviour	19%	9%	5%
■ Attitude	31%	28%	26%
■ Motivation	47%	24%	48%
■ Job Satisfaction	41%	31%	53%
■ Performance	21%	3%	12%

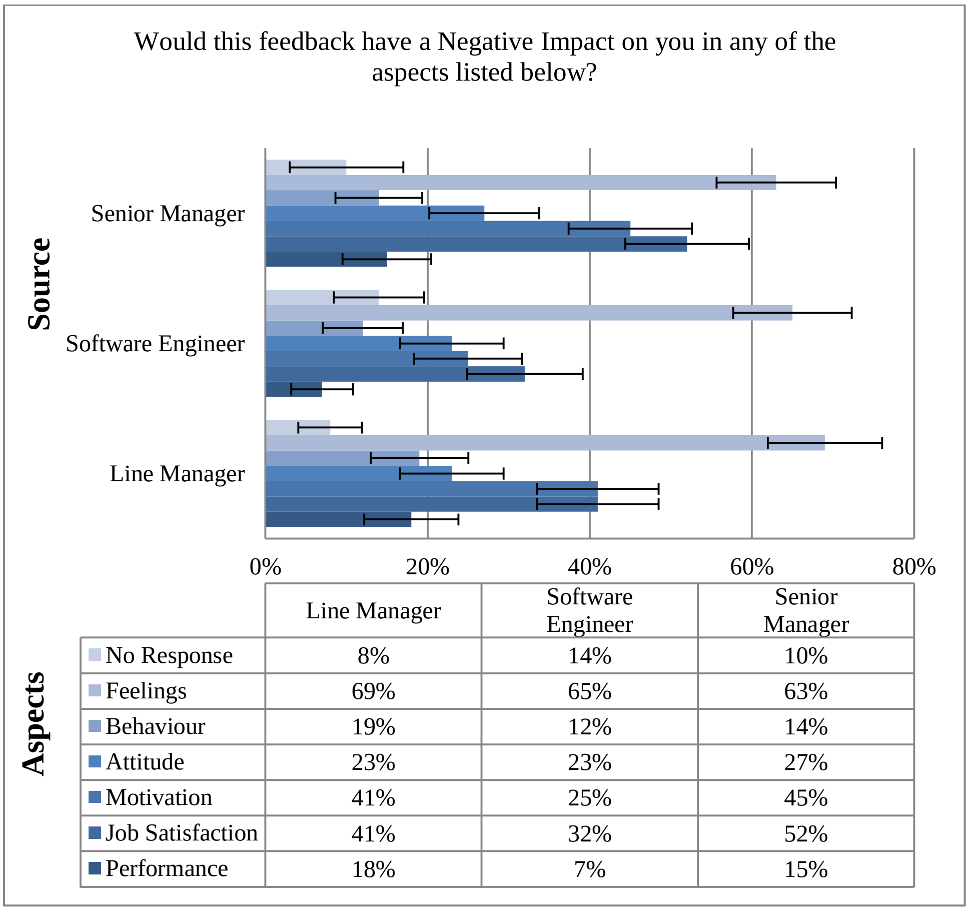
FIGURE 7.12 – INVESTIGATION 1, QUESTIONS 10-12: SUB-GROUP REPRESENTATION

Investigation 2: Participants who reported equal or higher value for feedback from an experienced or senior software engineer compared to both a line manager and senior manager. 76% (119/157) of the participants were included in this sub-group. The results of the impact reported by the participants in response to questions 7 to 12 is shown in Figure 7.13 and Figure 7.14. These findings are discussed and used in section 7.6.1 to support or dispute the research predictions presented in section 7.1.3.



	Line Manager	Software Engineer	Senior Manager
No Response	1%	1%	3%
Feelings	77%	76%	66%
Behaviour	24%	21%	18%
Attitude	51%	45%	42%
Motivation	80%	61%	69%
Job Satisfaction	91%	68%	82%
Performance	43%	33%	31%

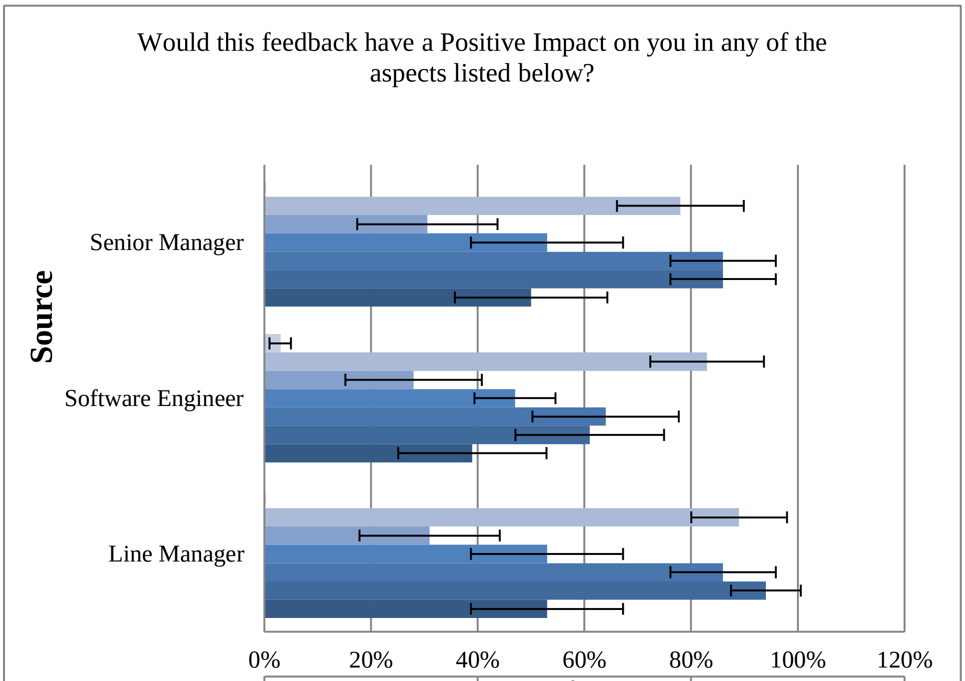
FIGURE 7.13 – INVESTIGATION 2, QUESTIONS 7-9: SUB-GROUP REPRESENTATION



	Line Manager	Software Engineer	Senior Manager
■ No Response	8%	14%	10%
■ Feelings	69%	65%	63%
■ Behaviour	19%	12%	14%
■ Attitude	23%	23%	27%
■ Motivation	41%	25%	45%
■ Job Satisfaction	41%	32%	52%
■ Performance	18%	7%	15%

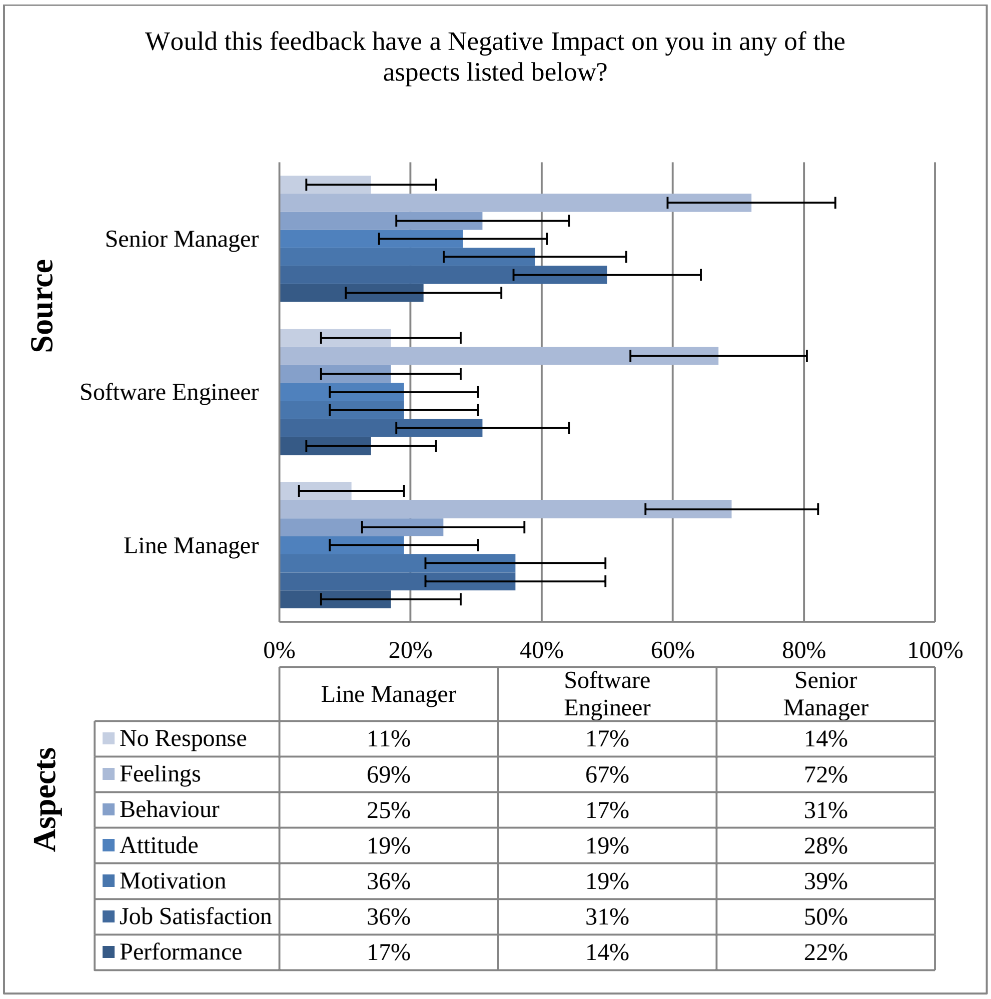
FIGURE 7.14 – INVESTIGATION 2, QUESTIONS 10-12: SUB-GROUP REPRESENTATION

Investigation 3: Participants who reported an equal value for feedback from an experienced or senior software engineer compared to both a line manager and senior manager. 23% (36/157) of the participants were included in this sub-group. The results of the impact reported by the participants in response to questions 7 to 12 is shown in Figure 7.15 and Figure 7.16. These findings are discussed and used in section 7.6.1 to support or dispute the research predictions presented in section 7.1.3.



	Line Manager	Software Engineer	Senior Manager
■ No Response	0%	3%	0%
■ Feelings	89%	83%	78%
■ Behaviour	31%	28%	30.56%
■ Attitude	53%	47%	53%
■ Motivation	86%	64%	86%
■ Job Satisfaction	94%	61%	86%
■ Performance	53%	39%	50%

FIGURE 7.15 – INVESTIGATION 3, QUESTIONS 7-9: SUB-GROUP REPRESENTATION

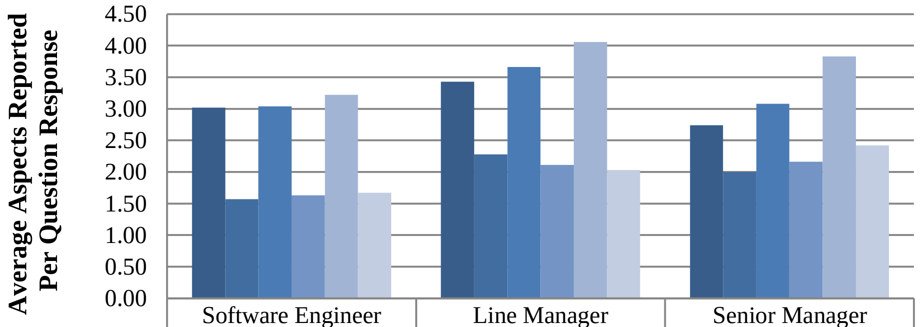


	Line Manager	Software Engineer	Senior Manager
■ No Response	11%	17%	14%
■ Feelings	69%	67%	72%
■ Behaviour	25%	17%	31%
■ Attitude	19%	19%	28%
■ Motivation	36%	19%	39%
■ Job Satisfaction	36%	31%	50%
■ Performance	17%	14%	22%

FIGURE 7.16 – INVESTIGATION 3, QUESTIONS 10-12: SUB-GROUP REPRESENTATION

Investigation 4: Participants who reported a less value for feedback from an experienced or senior software engineer compared to both a line manager and senior manager. 8% (13/157) of the participants were included in this sub-group. Due to the small representation of the participants, the results of this investigation are not included. These findings are discussed and used in section 7.6.1 to support or dispute the research predictions presented in section 7.1.3.

Figure 7.17 displays the mean number of aspects (standard deviation: Table 7.10) reported by each participant for each question as being affected by feedback. Figure 7.17 shows that the number of aspects reported by the participants as being affected varied in relation to the content polarity and the source of the feedback. In Figure 7.17, the key of I1, I2, or I3 relates to the sub-group investigation of participants, followed by the relevant question number.



Sub-Groups	Software Engineer	Line Manager	Senior Manager
■ I1 - Q7-9	3.02	3.43	2.74
■ I1 - Q10-12	1.57	2.28	2.00
■ I2 - Q7-9	3.04	3.66	3.08
■ I2 - Q10-12	1.63	2.11	2.16
■ I3 - Q7-9	3.22	4.06	3.83
■ I3 - Q10-12	1.67	2.03	2.42

FIGURE 7.17 – MEAN ASPECTS IMPACTED BY FEEDBACK RECEIVED FROM DIFFERENT SOURCES

TABLE 7.10 – MEAN AND STANDARD DEVIATION OF ASPECTS IMPACTED BY FEEDBACK RECEIVED FROM DIFFERENT SOURCES

	I1 - Q7-9	**I1 - Q7-9 StdDev**	**I1 - Q10-12**	**I1 - Q10-12 StdDev**
Software Engineer	3.02	1.32	1.57	1.11
Line Manager	3.43	1.5	2.28	1.58
Senior Manager	2.74	1.43	2.00	1.4
	I2 - Q7-9	**I2 - Q7-9 StdDev**	**I2 - Q10-12**	**I2 - Q10-12 StdDev**
Software Engineer	3.04	1.43	1.63	1.21
Line Manager	3.66	1.56	2.11	1.51
Senior Manager	3.08	1.52	2.16	1.59
	I3 - Q7-9	**I3 - Q7-9 StdDev**	**I3 - Q10-12**	**I3 - Q10-12 StdDev**
Software Engineer	3.22	1.61	1.67	1.49
Line Manager	4.06	1.47	2.03	1.65
Senior Manager	3.83	1.58	2.42	2.02

7.4.2 POST-SURVEY INTERVIEW FINDINGS

Of the 157 participants, 27 indicated at the time of completing the study that they would be willing to be contacted for a follow-up interview. Of the 27 contacted, interviews were arranged and completed with 14 software engineers, with 13 completed using Skype, and 1 completed face-to-face at the Open University's Milton Keynes campus.

The participants discussed their responses to the online survey in a structured interview. The questions were intended to investigate the participants' interpretation of terminology used in the survey. From the interviews, the participants indicated that their understanding of the terms used in the interview was as intended in the design of the study.

7.5 THREATS TO VALIDITY

The findings of this study are subject to three main threats to validity. First, as the sample is self-selected, the participants may not be a representative sample of software engineers. This provides a threat to the external validity of the findings. These findings should be viewed as indicative, and not as representative of all software engineers.

Second, this study was built on the findings of the previous studies presented in this thesis, and provided the participants with options in response to 12 scenarios and questions that were found in the previous empirical studies. This poses a potential threat to the construct validity of the findings, because it relies on the accuracy and representativeness of the previous studies, and limits the participants in this study from providing new ideas and responses to the scenarios presented. To reduce this threat to validity, the participants were provided with free-text boxes during the survey if they wished to provide an alternative response.

Third, the findings of this study are limited by the findings of the earlier studies. The earlier studies informed this study, directing the focus away from personal relationships and towards hierarchical and technical respect. While this provides a threat to the construct validity of the findings, the threat is limited by the corroborating evidence collected in the earlier studies. The earlier studies were underpinned by research method triangulation that supports the findings of the studies and provided confidence in the focus and scope of the design of this study.

7.6 SUMMARY

This study focused on the impact of a change in value for the feedback characteristics medium and source. These two feedback characteristics were identified in previous studies (chapters 5 and 6) as being key factors that can affect the impact of feedback on the motivation of software engineers. The findings of this study illustrate the differing opinions on the impact of a change in the source and medium feedback characteristics.

The participants. 157 participants successfully completed the survey. The survey was intended for active software engineers who consider themselves first and foremost software engineers, who work in a team environment, and who often write new code. This description was included in all invitations.

The medium. The participants reported different preferences for how they would prefer to receive feedback. Of the five options presented to the participants for receiving feedback, “notice board” was found to be the least popular medium for receiving feedback. The results indicate that the participants are more flexible concerning the medium used by their line manager and senior managers, with over 50% of the participants indicated two different media. Alternatively, feedback from a software engineer was expected to be received during a casual chat, with the second most commonly reported medium being reported by less than 40% of the participants.

When asked about the impact of a receiving feedback from a non-preferred medium, over 50% of participants said this would change the impact of the feedback if it came from their line manager or a senior manager, and 42% said it would change the impact if the source of the feedback was a software engineer. When asked to rate the change in the impact from minor to moderate when receiving feedback through a non-preferred medium, less than 20% of the participants indicated that the change in impact of the feedback would be ‘major’, with 80% indicating a ‘moderate’ or ‘minor’ change to the impact of the feedback.

The source. The participants were asked to respond to 9 different scenarios and questions that changed the source of the feedback to investigate what impact the source would have on the participants. The questions investigated how the participants ‘valued’ feedback from a source, and how specific types of feedback, praise and critical comments, would impact the participant. The participants were also presented with scenarios of when they would receive feedback and were asked to indicate if this feedback would impact any of the 6 listed aspects (performance, job satisfaction, motivation, attitude, behaviour, and feelings).

Over 50% of the participants indicated that feedback from a senior or experienced software engineer was more valuable to them than feedback from a senior manager, and over 80% of the participants indicated that feedback from a senior or experienced software engineer was equal or more valuable to them than feedback from their line manager. 55% of the participants reported that feedback from the company CEO was as valuable as feedback from a senior manager, and over 55% indicated that feedback from a line manager was equally as valuable as feedback from a senior manager. The results suggest that the least valuable source of feedback is new software engineers, and that software engineers with both knowledge and experience are the most valuable source of feedback.

When asked to report the impact of praise or critical comments, the responses of the participants changed in comparison to the findings from the feedback value question. Praise and critical comments from new software engineers, like feedback value, were reported to have the lowest impact, but feedback from a line manager or senior manager became more important in comparison to responses to the feedback value question. This suggests that, while people may 'value' feedback from their colleagues more, the impact of feedback received from someone above the participant in their company's hierarchy is greater than the value of the sources feedback may suggest.

During the analysis sub-groups of participants were identified and their findings investigated. The first investigation identified 38 participants who reported that feedback from experienced and senior software engineers was more valuable than feedback from their line manager or a senior manager. The same 38 participants indicated that more aspects were affected by feedback received from their line manager than from another software engineer. Further comparisons focused on the same relationship using a 'greater or equal' comparison, and found the same results: the feedback that affected the most aspects was feedback received from a line manager. The only time feedback from a software engineer was reported to impact more aspects was when compared to senior managers, in all other findings, feedback from software engineers was reported as impacting the least number of aspects of all of the three sources.

7.6.1 PREDICTION COMPARISON

During the research design phase a set of eight predictions were identified and made prior to the collection of data. These predictions were formed from the findings of the previous studies, and intended to use findings from previous studies to anticipate the some of the findings of this study. In the following paragraphs, each prediction will be evaluated and compared to the findings of this study.

Prediction 1 – Typically, participants will report the value of feedback from a 'new software engineer' as of less value than the value of feedback from an 'experienced software engineer' or a 'senior software engineer'.

> **Prediction 1 supported.** Over 60% of the participants reported more value for feedback from an experienced or senior software engineer more than feedback from a new software engineer, and over 98% of the participants reported that feedback from an experienced or senior software engineer was of equal or greater value than feedback received from a new software engineer.

Prediction 2 – Positive feedback will affect job satisfaction for the majority of respondents.

> **Prediction 2 supported.** Over 80% of the participants reported that positive feedback would impact job satisfaction. This number ranged depending on the feedback source, with feedback from a line manager impacting job satisfaction for 90% of the participants, feedback from a senior manager impacting job satisfaction for 82% of the participants, and feedback from a software engineer impacting job satisfaction for 66% of the participants. The confidence interval range of 86-94%, 77-87%, and 60-72% respectively, increases the likelihood that this finding is not due to coincidence.

Prediction 3 – Negative feedback will affect feelings for the majority of respondents.

> **Prediction 3 supported.** Feelings were reported as being impacted by over 60% of the participants. Feelings were found to be impacted by feedback from a line manager (69%), senior manager (63%), and software engineer (64%). The confidence interval range of 63-75%, 57-69%, and 58-70% respectively, increases the likelihood that this finding is not due to coincidence.

Prediction 4 – Negative feedback will cause the majority of respondents to want to fix the problem.

> **Prediction 4 supported.** Over 80% of the participants reported that their reaction to receiving negative feedback would be to want to fix the problem. The number of participants who reported this response depended on the source

of the feedback, with 85% of participants wanting to fix the problem in reaction to receiving negative feedback from a line manager, 78% of participants wanting to fix the problem in reaction to receiving negative feedback from a senior manager, and 78% of participants wanting to fix the problem in reaction to receiving negative feedback from a software engineer. The confidence interval range of 80-90%, 73-83%, and 73-83% respectively, increases the likelihood that this finding is not due to coincidence.

Prediction 5 – If a change in medium is reported as changing the impact of the feedback, it will rarely be a major change.

Prediction 5 supported. Half of the participants reported that a change in medium would alter the impact of feedback, and of those participants, less than 20% reported the change would be major. The number of participants that indicated that a change in medium would change the impact of received feedback ranged: 52% (line manager), 52% (senior manager), and 42% (software engineer). Of those participants, 16% indicated a major change for feedback received from a line manager, 19% indicated a major change for feedback received from a senior manager, and 9% indicated a major change for feedback received from a software engineer. The confidence interval range of 45-57%, 45-57%, and 36-48% respectively, increases the likelihood that this finding is not due to coincidence.

Prediction 6 – If a participant scores feedback from software engineers as having greater value than feedback from managers in question 4, then question 5 and 6 will show a similar set of responses.

Prediction 6 NOT supported. 37% of the participants scored feedback from senior or experienced software engineers as of greater value than feedback from both a line manager and senior managers, but only 9% of total participants (24% of the subset of participants) continued this pattern in their responses to questions 5 and 6.

Prediction 7 – If a participant scores feedback from software engineers as of equal or greater value than feedback from the managers in question 4, then questions 7-12 will show a greater level of impact (more aspects selected) for the software engineer feedback.

Prediction 7 NOT supported. 76% of participants scored feedback from senior or experienced software engineers as of equal or greater valuable than feedback from both a line manager and senior managers, but reported less impact from feedback received from a software engineer. Positive feedback from a software engineer impacted 3.04 aspects per person, compared to positive feedback from a line manager impacting 3.66 aspects per person. Negative feedback from a software engineer impacted 1.63 aspects per person, compared to 2.11 aspects per person from negative feedback received from a line manager.

Prediction 8 – If a participant scores feedback from managers as of greater value than feedback from software engineers in question 4, then questions 7-12 will show a greater level of impact (more aspects selected) for the manager feedback.

Prediction 8 NOT supported. There was not enough data provided to confirm or refute this prediction, as less than 10% of the participants reported feedback from senior managers and line manager as more valuable than feedback from experienced or senior software engineers.

7.7 SO WHAT DO WE KNOW NOOW?

From the findings identified over multiple studies presented in chapters 5, 6 and 7 we now have a much more complete picture of feedback and its impact in software engineering environments. Feedback has been identified and defined within the context of software engineering, important characteristics have been discovered and investigated, and data investigating the impact of a change in value for three feedback characteristics (medium, source, and content polarity) has been collected. The next paragraphs will briefly summarise the findings of this research, prior to the complete discussion that follows in chapter 9.

Feedback. Ten feedback characteristics, identified from the findings of a previous study (chapter 3), and investigated during this research (chapters 3, 5, 6 and 7), are used by software engineers to describe feedback. Feedback in software engineering should not be viewed as a single construct, but as the combination and interaction of ten characteristics.

Feedback characteristics. The characteristics identified include the medium, the source, and the content polarity, which were investigated during this study. Several studies (chapters 5, 6, and 7) have identified that the values of the source, medium, and content polarity can alter the effect of received feedback.

Feedback impact. The findings presented in this study support earlier findings (chapters 5 and 6) that positive feedback affects job satisfaction, and that negative feedback leads to the recipient wanting to fix the problem. Both negative and positive feedback were reported to affect other aspects, including motivation, attitude, behaviour, performance, and feelings.

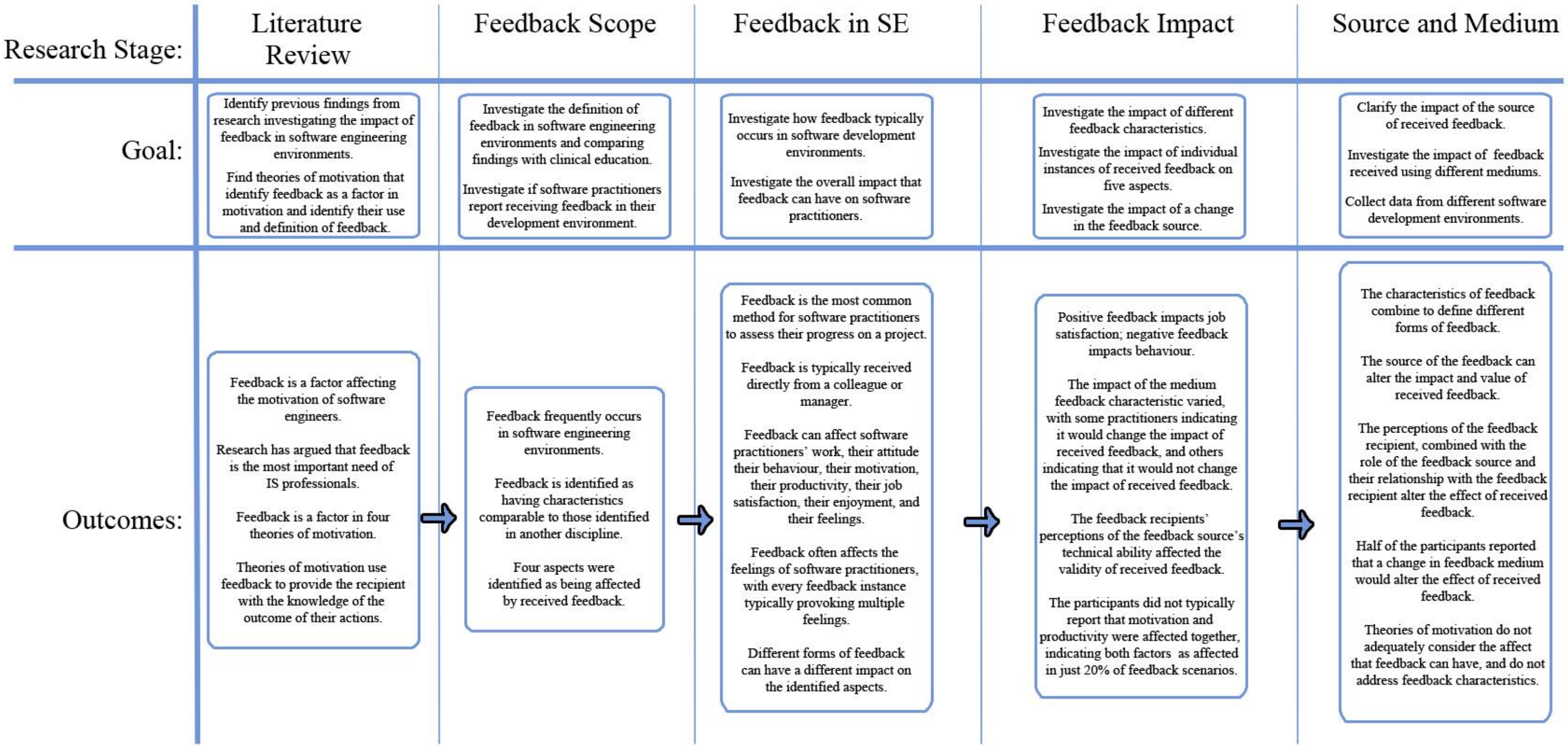
Feelings. Throughout these findings and often overshadowed, is the affect that feedback is reported to have on the feelings of software engineers. Both negative and positive feedback were identified as affecting the recipients feelings and all the participants at every empirical study during this research identified feelings as being impacted at some point. The emotional reaction to receiving feedback requires further investigation.

The source. The source affected the impact of feedback for the majority of software engineers. Often the participants reported valuing feedback more or less depending on the source of the feedback. While some participants indicated feedback as more valuable from their fellow software engineers, this did not relate to the impact of the feedback. Participants who valued feedback from a software engineer more than feedback from other sources, reported that feedback from managers would affect more aspects than feedback from a software engineer.

The medium. The participants reported different preferences for receiving feedback, and half of the participants in this study reported that receiving feedback via a non-preferred medium would alter the impact of the feedback. While the impact of a change in the medium did not typically lead to a major change in the impact of the feedback, a significant number of software engineers reported the medium as having an effect on the impact of received feedback.

Hence, feedback is an influential factor in software engineering environments, and can affect software engineers’ motivation, attitude, behaviour, job satisfaction, performance, and feelings. The impact and the value of feedback can be different for each software engineer, suggesting that any source of feedback may consider providing feedback that is suitable to the feedback recipient.

An updated map of the research progress is shown in Figure 7.18.



Research Stage:	Literature Review	Feedback Scope	Feedback in SE	Feedback Impact	Source and Medium
Goal:	Identify previous findings from research investigating the impact of feedback in software engineering environments. Find theories of motivation that identify feedback as a factor in motivation and identify their use and definition of feedback.	Investigate the definition of feedback in software engineering environments and comparing findings with clinical education. Investigate if software practitioners report receiving feedback in their development environment.	Investigate how feedback typically occurs in software development environments. Investigate the overall impact that feedback can have on software practitioners.	Investigate the impact of different feedback characteristics. Investigate the impact of individual instances of received feedback on five aspects. Investigate the impact of a change in the feedback source.	Clarify the impact of the source of received feedback. Investigate the impact of feedback received using different mediums. Collect data from different software development environments.
Outcomes:	Feedback is a factor affecting the motivation of software engineers. Research has argued that feedback is the most important need of IS professionals. Feedback is a factor in four theories of motivation. Theories of motivation use feedback to provide the recipient with the knowledge of the outcome of their actions.	Feedback frequently occurs in software engineering environments. Feedback is identified as having characteristics comparable to those identified in another discipline. Four aspects were identified as being affected by received feedback.	Feedback is the most common method for software practitioners to assess their progress on a project. Feedback is typically received directly from a colleague or manager. Feedback can affect software practitioners’ work, their attitude their behaviour, their motivation, their productivity, their job satisfaction, their enjoyment, and their feelings. Feedback often affects the feelings of software practitioners, with every feedback instance typically provoking multiple feelings. Different forms of feedback can have a different impact on the identified aspects.	Positive feedback impacts job satisfaction; negative feedback impacts behaviour. The impact of the medium feedback characteristic varied, with some practitioners indicating it would change the impact of received feedback, and others indicating that it would not change the impact of received feedback. The feedback recipients’ perceptions of the feedback source’s technical ability affected the validity of received feedback. The participants did not typically report that motivation and productivity were affected together, indicating both factors as affected in just 20% of feedback scenarios.	The characteristics of feedback combine to define different forms of feedback. The source of the feedback can alter the impact and value of received feedback. The perceptions of the feedback recipient, combined with the role of the feedback source and their relationship with the feedback recipient alter the effect of received feedback. Half of the participants reported that a change in feedback medium would alter the effect of received feedback. Theories of motivation do not adequately consider the affect that feedback can have, and do not address feedback characteristics.

Research Stage Key:

Literature - Literature Review, chapter 2
Feedback Scope - Feedback Scoping Study, chapter 3
Feedback in SE - Feedback in One Software Engineering Environment, chapter 5
Feedback Impact - The Short-Term Impact of Received Feedback, chapter 6
Source and Medium - The Effect of ‘Source’ and ‘Medium’ Feedback Characteristics, chapter 7

FIGURE 7.18 – UPDATED RESEARCH PROGRESS MAP

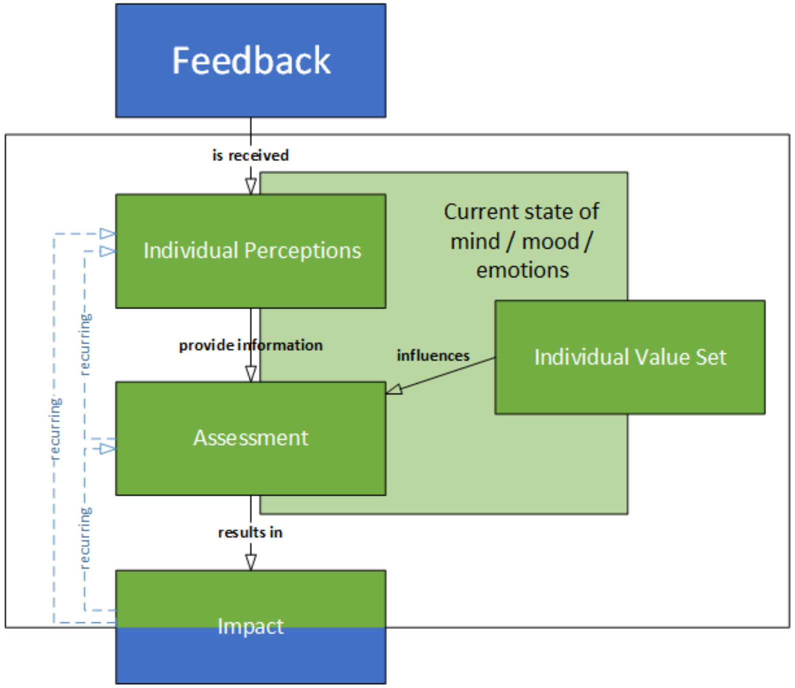
CHAPTER 8

A MODEL OF FEEDBACK IN SOFTWARE ENGINEERING

This chapter presents a model capturing what happens when feedback is received in software engineering environments. The model combines knowledge found in the literature with the findings of this research to provide a discipline-specific representation. The model is presented, research underpinning the model is identified, and example instances of feedback are used to illustrate the different stages of the model.

The empirical research presented in this thesis identified feedback as a way in which progress is evaluated in software engineering environments, defined by characteristics comparable to those identified in clinical education (van de Ridder *et al.*, 2008), and software engineers reported that the effect of feedback depended on the values of the characteristics of received feedback. Software engineers reported a range of different preferences and values which they use to evaluate any feedback they receive, establishing the feedback's worth and/or importance to them, which subsequently results in the impact that the received feedback will have.

The following sections present and discuss a model (8.1), discuss the foundations of the model (8.2), present example feedback instances compared to the model (8.3), and present a summary of the model (8.4).



Feedback

Feedback Characteristics:
- Content
- Direction
- Goal
- Instigation
- Medium
- Preparation
- Recipient
- Setting
- Source
- Timeliness

Current state of mind / mood / emotions

An individual's current state of mind / mood / emotions can affect their perceptions, their assessments, and their individual value set

Individual Perceptions

The characteristics of feedback as interpreted by the individual's perception, attention and understanding; the subjective view of the received feedback

Impact

Affected Aspects:
- Attitude
- Behaviour
- Feelings
- Job Satisfaction
- Motivation
- Performance

Individual Value Set

Identified in Software Engineering:
Feedback Preferences:
- Preference for the medium
- Preference for the setting

Feedback Value:
- Career progression feedback
- Technical feedback

Source Value:
- Hierarchy, or
- Technical Respect, or
- Hierarchy and Technical Respect , or
- No source-specific value

Assessment

Goal:
- Acceptable goal
- Unacceptable goal

Hygiene Factors:
- Implications for career progression
- Implications for current role
- Implications for work life

Preference:
- Of the medium
- Of the setting

Validity:
- Of the feedback content
- Of the feedback source

Value:
- Of the feedback content
- Of the feedback source

FIGURE 8.1 – A MODEL OF FEEDBACK IN SOFTWARE ENGINEERING

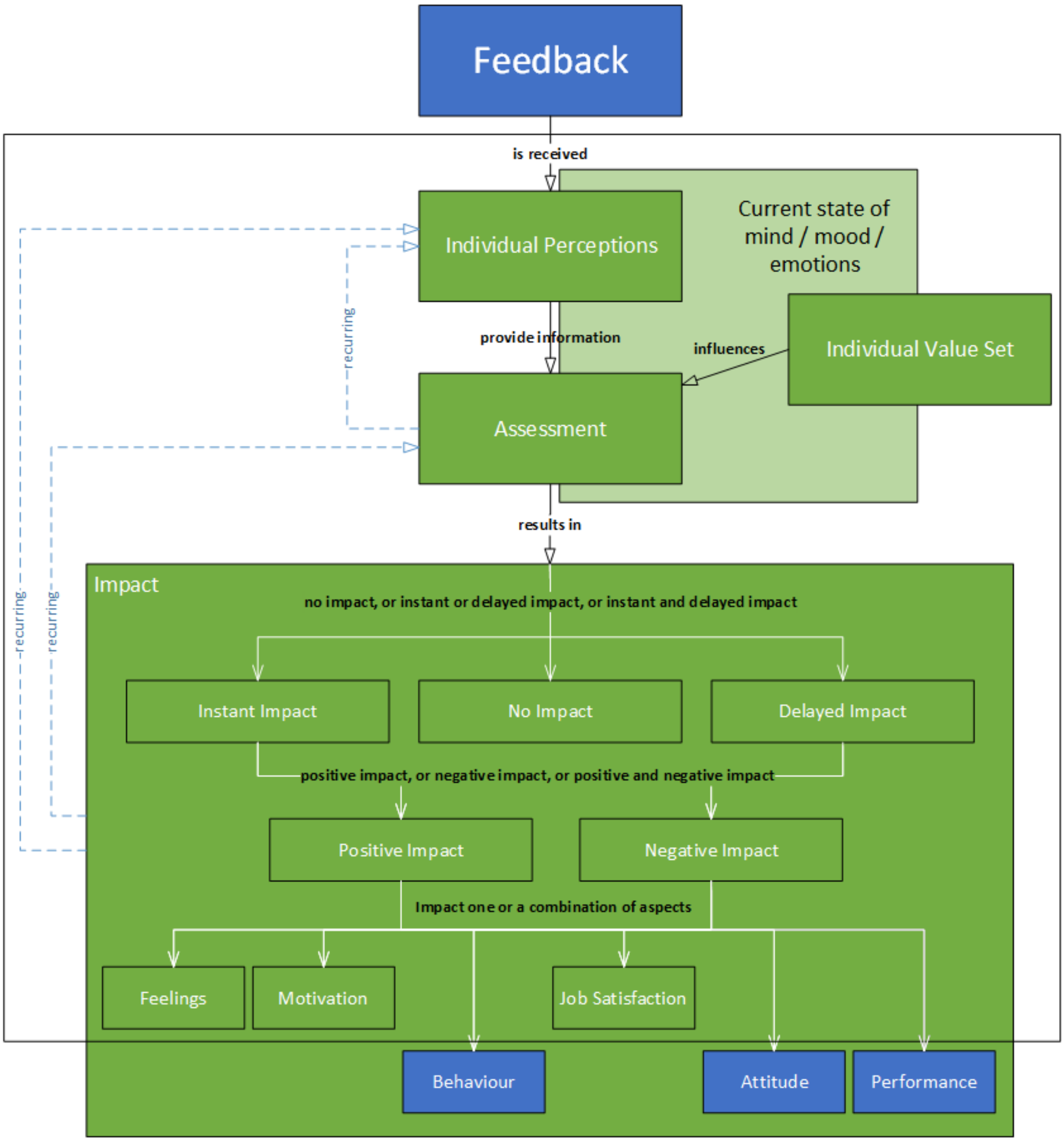


FIGURE 8.2 – A DETAILED MODEL OF FEEDBACK IN SOFTWARE ENGINEERING

8.1 A MODEL

A model of feedback in software engineering is shown in Figure 8.1 and Figure 8.2. Figure 8.1 provides an overview of the model, and Figure 8.2 represents the impact of received feedback in more detail. This model captures what happens when feedback is received, including the perceptions, assessments and impacts as a result of receiving feedback. The model emerges as a combination of both the literature and the findings of studies with software engineers presented in chapters 3, 5, 6 and 7. The model is an information processing system, as seen in cognitive science literature (examples: Newell and Herbert (1972), Feder and Kelly (1986), Broadbent (1958)), where stimulus (feedback) provides the input, the input is processed (individual perceptions, individual *value set*, assessments), and a response (impact) is the output.

In short, when feedback is received, the individual perceives the values of the feedback characteristics. These perceptions provide information for the individual, and the individual's *value set* influences several assessments related to the value and validity of the feedback. The individual's perceptions, assessments and *value set* are all influenced by the individual's current state of mind / mood / emotions. The assessments about the feedback's value and validity result in the impact, or no impact, of the received feedback. This process appears linear, but it is important to note that while the process of perceptions, assessments, and impact is linear, the overall process of receiving feedback is not. The feedback recipient will repeat the different stages in the model multiple times for each instance of received feedback, and possibly at several different occasions after the feedback has been received.

In both Figure 8.1 and Figure 8.2, the different sections of the model that are internal and external for the individual are addressed by colour. As the impact of received feedback can be both internal and external, it is represented by a split-colour box in Figure 8.1. Blue represents an external facet of the model, such as the feedback that the participant receives, or eventually the impact the received feedback may have on their behaviour or performance. Green represents an internal facet of the model, such as the individual's perceptions, their assessment, or the eventual impact the received feedback may have on their feelings or motivation. A transparent box with a black border also indicates the separation of internal and external facets in Figure 8.1 and Figure 8.2.

In the following sub-sections, further detail will be presented on the different areas of the model, identifying the different stages of the model and how the model works.

8.1.1 INDIVIDUAL PERCEPTIONS

As presented and discussed at various stages within this research (chapter 3, chapter 5, chapter 6, chapter 7), software engineers discussed several characteristics of feedback. The values of feedback characteristics are used to distinguish between different forms of feedback. When a software engineer receives feedback, the individual will perceive the value of one or multiple characteristics of the received feedback.

The feedback characteristics are listed in Figure 8.1. In addition to perceiving the values of feedback characteristics, the feedback recipient also perceives specific information about the source and the content.

- **The Source**
  - Experience
  - Knowledge
  - Common working history
  - Expertise
  - Hierarchical position
  - Influence / power
- **The Content**
  - Topic
  - Polarity

The source and content have additional attributes perceived by the recipient that require clear identification. In this model, there is no identification of more or less important feedback characteristics.

These perceptions of the values of the received feedback characteristics provide the recipient with knowledge that will be used to aid assessments. It should be noted that as these are perceptions, each individual’s perception of the values of the characteristics of feedback may be different. It is possible that the perceptions of the values of the feedback characteristics will differ between the feedback source and the feedback recipient.

The perceptions of each individual can be influenced by their current state of mind, their mood or their emotions. Bouhuys *et al.* investigating the impact of an individual's affective state and found that participants who were induced to feel depressed reported ambiguous facial expressions as sadder than participants induced to feel elated (Bouhuys *et al.*, 1995). Research has suggested that extreme emotional stress can impair memory (Packard *et al.*, 1994), and an individual's perception of a task, and their global satisfaction with the task were found to be influenced by their affective state (Kraiger *et al.*, 1989). More recent studies have found that an individual's affective state can influence memory (Storbeck and Clore, 2005), attention (Phelps *et al.*, 2006) and their experience of the world (Riener *et al.*, 2011). Zeelenberg *et al.* (2006) found that participants were able to faster process emotionally significant stimuli (happy faces, pictures of mutilations, words such as death and love). Previous research investigating the impact of emotions or mood on receiving feedback was not found, but the findings from studies of an individual's affective state suggest that each individual's current state of mind, their mood and their emotions may alter how they perceive the characteristics of received feedback.

8.1.2 INDIVIDUAL VALUE SET

The findings presented in chapters 5, 6 and 7 found that software engineers' individual *value set* identified their focus within software engineering, their values regarding feedback content and feedback source, and their preferences when receiving feedback in software engineering environments. The individual *value set* is shown below:

- Source value:
  - Software engineers reported valuing feedback from several different types of source:
    - Sources for which they have technical respect.
    - Sources that are above them in the company's hierarchy.
  - The value attributed to the type of source varied for each participant.
  - Some engineers displayed no source-specific value.
- Feedback value:
  - Each software engineer reported valuing feedback of two distinct types:

        - Technical feedback.
        - Feedback related to career progress.
    - The value given to each type of feedback varied for each participant.
- Feedback preferences:
    - Software engineers reported preferences for the medium of received feedback:
        - Some engineers preferred to receive feedback via a specific media.
        - Some engineers reported no preference relating to the medium.
    - Software engineers reported preferences for the setting of received feedback:
        - Some engineers preferred to receive feedback in a specific setting.
        - Some engineers reported no preference relating to the setting.

The individual's *value set* influences the individual's assessment of the received feedback, and each individual uses the information from the individual's perceptions and individual's *value set* to make assessments on the value of received feedback. Other individual *value set* factors may also influence subsequent assessments, and other individual *value set* factors may not be specific to software engineering environments. The model accounts for all factors found in the reviewed literature and identified during this research, but there may be other undiscovered factors.

8.1.3 ASSESSMENTS

After the feedback recipient has perceived the values of the feedback characteristics, their individual *value set* influences the individual's assessments of the value and validity of the received feedback. Data presented in chapters 5, 6 and 7 explored the effect of different feedback characteristics, and identified that there was no consistent relationship between the values of the characteristics of received feedback and the reported impact. This finding was attributed to an individual *value set*. An individual's *value set* influences the assessments made on the value of the received feedback.

Feedback value and validity is considered in two areas:

- Feedback content.
- Feedback source.

Three other factors are also considered:

- Individual preferences, for:
  - Medium.
  - Setting.
- The goal of the feedback source
- Hygiene considerations.

In the following paragraphs, the five identified factors are discussed in greater detail.

Feedback content is the information contained within the feedback – what it is about (performance, attitude, behaviour, recent work, past project etc.), and the polarity (positive, negative, neutral). The feedback recipient has already perceived this information, and now must assess it based on the individual's characteristics if they believe it is both valid, and valuable. The validity of the feedback is based on the individual's perceptions of the feedback content, and whether the individual believes the feedback to be a truthful account. The value of the feedback is distinctly different, and focuses on whether the recipient values the content of the feedback. For example, feedback on a participant's attitude may be valid, but the participant may not value this form of feedback.

Feedback source includes all of the attributes of the source as identified during individual perceptions. The participant perceives the source's experience, their knowledge, their common working history, their expertise, their hierarchical position, and their influence and power. These perceptions combined with the recipient's individual *value set* help decide the validity and the value of the feedback source. The feedback recipient will determine if the source of the feedback is valid, relating specifically to the feedback they are given. For example, highly technical feedback from a non-technical project manager may be considered invalid due to the source's knowledge and expertise, or lack of it. The recipient will then consider the value of the received feedback, relating to their individual *value set*. For example, if an individual only values technical knowledge, it is unlikely that feedback from a non-technical source will be of value.

Individual preferences are the specific likes and dislikes of the feedback recipient that are not clearly defined within the other factors. These include preferences to the setting in which feedback is received (examples: casual chat, team meeting, one to one meeting) and the medium used to send the feedback (examples: email, face to face, notice board). The findings presented in chapters 6 and 7 identified that, while individual preferences will not directly remove all value from received feedback, individual preferences can have an effect on the impact of received feedback.

Feedback goal is the recipient's perception of the intention behind the feedback. The goal of the feedback would typically come from the feedback source, but it is possible for the goal of the feedback to come from another source. For example, a senior manager may instruct a manager to provide feedback to a software engineer with the goal of boosting the software engineer's confidence. The feedback source would be the manager, but the goal would have come from the senior manager. The feedback recipient may consider the feedback goal to be acceptable or unacceptable, changing the effect of the received feedback. If the goal of the feedback is judged to be acceptable, it will have no effect on the impact of the received feedback. However, if the recipient judges the feedback goal to be unacceptable, the impact of the received feedback will change. For example, if the feedback recipient receives feedback that could be described as praise for their recent performance, this will typically have a positive impact on their job satisfaction, but if they perceive the intention of the feedback to be unacceptable, it may change the impact that the received feedback has on the recipient, and may cause a negative impact on the recipient's job satisfaction.

Hygiene consideration is the recipient's need to maintain their work life. While a software engineer may prefer and value feedback more from a fellow software engineer, receiving negative feedback from a senior manager may have a strong effect because of the position and power of the source, and the implications the received feedback may have for the engineer's career, current role, and promotion prospects. The findings presented in chapter 7 identified that feedback that is considered the most valuable is not the same as feedback that has the most impact. This was found when respondents reported valuing feedback from software engineers above that from managers, but then reported a more aspects as being impacted by feedback received from managers. This suggests that while feedback from software engineers may be more valuable, feedback from managers who can influence their career, current role, and promotion prospects will have a more wide-spread impact on six aspects of software engineers.

During the assessment stage, software engineers may consider multiple factors at the same time. For example, technical feedback may be both valid and valuable for the recipient, but if the source has neither technical knowledge nor technical ability, feedback may be considered to be invalid. Equally, the feedback and the source may be considered both valid and valuable, but if the perceived goal of the feedback is not acceptable, the feedback may have a different impact than if the goal of the feedback were considered acceptable. In summary, the assessment process is a combination of multiple assessments to decide the overall impact of the received feedback.

8.1.4 IMPACT

After the values of the feedback characteristics have been perceived, and the information collected has been used to inform assessments that are influenced by an individual's *value set*, the impact of the received feedback can be established. Overall, impact falls into three categories: immediate impact, delayed impact, and no impact. It is possible that any one cycle of perception-knowledge-assessment-impact will have a delayed impact, an immediate impact, both a delayed and an immediate impact, or no impact. For anything other than no impact, the impact can then be described as positive or negative, as considered by the feedback recipient. In the following paragraphs, the different types of impact are discussed in further detail.

No impact is when the received feedback does not have an impact on the feedback recipient. This typically occurs when the feedback is considered to be invalid, either by the source or the content of the feedback. However, invalid feedback does not always mean that there will be no impact.

Immediate impact is when the received feedback has an impact on the participant at the time of receiving the feedback, and could also be described as the initial reaction to receiving feedback.

Delayed impact is when the received feedback has an impact on the participant after the time of receiving the feedback. To distinguish between an instant and a delayed impact, if the impact occurs when the participant first receives the feedback, it is an immediate impact, and if the impact occurs at a later time or date, it is a delayed impact.

Positive impact is when the recipient considers the impact to be favourable, such as increasing their motivation, or making them feel happy or proud. The positive impact can be immediate, delayed, or both.

Negative impact is when the recipient considers the impact to be detrimental, such as reducing their job satisfaction, or lowering their performance. The negative impact can be immediate, delayed, or both

Affected aspects are the six aspects reported both in the literature and in the findings of empirical studies in this research (chapters 3, 5, 6 and 7) as being affected by received feedback: feelings, motivation, behaviour, job satisfaction, attitude, and performance. These aspects can be affected positively or negatively, and can be affected individually or in combination together, as either an instant or delayed impact.

Overall, the impact of received feedback is a combination of the three parameters:

1. Instant, delayed, or no impact.
2. Positive or negative impact.
3. Impact on feelings, motivation, behaviour, job satisfaction, attitude, or performance.

For example, the impact of received feedback could be described as *an instant positive impact on feelings* or as *a delayed negative impact on performance* or as *no impact*. Additionally, the impact can be a combination of both positive and negative and can affect one or a combination of the aspects.

8.1.5 MODEL SUMMARY

The model identifies the perception-knowledge-assessment-impact cycle software engineers experience when they receive feedback, and untangles some of the causality behind the different impact of received feedback for different software engineers. An overall view of the process is presented, and a more detailed view of the impact is provided.

While the model appears linear, it is a recurring cycle that may happen many times for any one instance of received feedback. Perceptions may change, altering assessments, and changing the impact of the received feedback. Software engineers may re-asses received feedback, and may re-perceive the characteristics of received feedback. The exact mechanism by which the impact of received feedback may alter with time is not yet understood, but the findings of this research suggest that participants reflect and consider received feedback, which may change the delayed impact of the feedback. While this reflection cannot alter the immediate impact of received feedback, the delayed impact can change.

8.2 MODEL FOUNDATION

This section presents the relevant data and literature that supports the presented model.

PRELIMINARY STUDY

The literature review (chapter 2) identified that the definition of feedback had been investigated in multiple disciplines, however it was acknowledged that “little empirical effort has been expended in an attempt to understand [feedback]” (Herold and Greller, 1977: 142) in management literature, and that “dynamic psychotherapy shows little understanding or use of the concept of feedback” (Berger, 1994: 235). However, van de Ridder *et al.* (2008) investigated feedback in clinical education, and reported that feedback was not a singular construct, but was the combination of several characteristics.

A study was conducted (chapter 3) that identified ten characteristics of feedback (see Figure 8.1) as reported by software engineers that were comparable to those identified by van de Ridder *et al.* (2008). These characteristics combine to define feedback, and identify different forms of feedback by the values of the characteristics. Further empirical studies (chapters 5, 6 and 7) used the identified feedback characteristics while investigating the impact of feedback in software engineering environments. The overall findings from the empirical studies in this thesis suggest that the ‘source’ and ‘content’ feedback characteristics have other sub-characteristics, as reported by software engineers. These characteristics are perceived by software engineers when they receive feedback.

The same study (chapter 3) investigated the impact of received feedback. The participants identified four aspects that were affected by received feedback: Motivation, behaviour, job satisfaction and feelings. These aspects are included in the model (Figure 8.2).

Chapter 8

PERSONAL PROFILES

The Feedback in One Software Engineering Environment (chapter 5) provided an overview of individual profiles compiled from data collected from each participant. These profiles were updated with data collected during the Short-Term Impact of Received Feedback Study (chapter 6). An analysis of these profiles led to the identification of two individual *value set* factors specific to software engineering environments (source value and feedback value). These characteristics were identified by evaluating the reasoning provided by the participants in their data, justifying why they reported that the impact of received feedback was different or changed dependent on the value of the characteristics of the feedback. For example, one participant reported that:

> “the bigger the cheese you impress the more motivated you get”.

This suggests that the participant values feedback from a source that is above him in the company’s hierarchy. This doesn’t mean that feedback from other sources is of no value to the participant. A different participant reported that he was:

> "more interested in my equals than I am project managers".

This suggests that the participant values feedback from his peers more than he does his project managers. This doesn’t mean feedback from project managers is of no value to the participant.

These identified characteristics specific to software engineering environments influence the assessments of source value and content value. The findings of the personal profiles suggest that each individual’s *value set* specific to software engineering environments will influence the decided value of the received feedback. However, it is not simply that if a software engineer values feedback from their project manager, that any feedback from their project manager will be considered valuable. There is a range of assessments that affect the value of feedback, and other individual *value set* factors must be considered.

Some software engineers reported a subtle preference for the setting of the received feedback. While often unremarked, this focused on social norms of not being given highly critical personal feedback in group situations, and, for some, extraordinary displays of praise in front of their peers were considered awkward and unnecessary. This was seen in comments expressing “that would usually come up in the one-to-one’s” and “they’ll just come over and tell me” and expressing that both of those situations were normal and expected depending on the form of feedback, such as positive comments from a project manager, or discussing bugs identified by a software tester.

SCENARIO RESPONSES

In the Short-Term Impact of Received Feedback Study (chapter 6) software engineers were asked to respond to seven scenarios in which they would receive feedback, and in a follow-up interview they were asked to discuss their responses and respond to changes to the seven scenarios. This study influenced the model by finding: (1) software engineers have preferences for the medium used to receive feedback, (2) feedback is more or less valuable depending on the source of the feedback, (3) feedback can be regarded as valid or invalid, and (4) feedback can cause a change in several aspects.

1. **Medium.** In one scenario the participants were asked if the impact of the received feedback would change if feedback sent by face to face communication, were actually sent by email. Of the sixteen respondents to this question, five indicated that the impact of the received feedback would not change, four indicated that the received feedback would have less impact, three indicated that the received feedback would have more impact, and three indicated that they preferred face-to-face feedback. These findings suggest that the medium can alter the impact of received feedback, but suggest that the impact is related to a personal preference regarding communication.
2. **Source.** In another scenario the participants were asked if the impact of the received feedback would change if feedback from their line manager, were instead received from their project manager. Of the fifteen respondents to this question, six indicated that it would not change the impact of the received feedback, four indicated that the feedback would have more impact, and three indicated that the feedback would have less impact. These findings suggest that some software engineers attribute more or less value to feedback depending on the source of the feedback.
3. **Validity.** Across several feedback scenarios, participants reported that sometimes the effect of received feedback would depend on the validity of

the feedback. One software engineer reported that feedback received from their project manager instead of their division head would have more impact because of it being “more believable” and that their project manager could “comment validly” on their work. Another participant questioned the validity of received feedback, reporting that the impact would depend on “why [line manager] thinks we should have progressed further”. These findings suggest that software engineers may attribute more or less value to feedback depending on if they believe the feedback is valid.

4. **Aspects.** When analysing the responses to the negative feedback scenarios, software engineers reported that the feedback would affect their behaviour, and that they would want to “fix the problem”. Attitude, job satisfaction, productivity, and motivation were also reported as being affected by received feedback. The model includes this finding by identifying behaviour, motivation, job satisfaction and attitude as aspects that can be affected by received feedback. Productivity is represented in the model by the performance aspect.

DIARY FEEDBACK INSTANCES

As presented in the Feedback in One Software Engineering Environment Study (chapter 5) , software engineers reported instances of feedback received during a one-week diary study. In the collected instances, software engineers reported the characteristics of received feedback, and indicated the impact the received feedback had on them. On average (mean) each instance of received feedback caused the participants to experience two or more different feelings. The impact on feelings influenced the model of feedback (Figure 8.2) to include feelings as an aspect that can be affected by received feedback.

DIARY DAY SUMMARIES

In the diary study (chapter 5), engineers recorded instances of feedback (discussed further in section 8.3) and day summaries: a summary of how each participant felt at the end of their working day and if the feedback they had received during their working day had affected them. During this study engineers reported that how they felt at the end of their working day could be affected by the feedback they received during the day, and that it could have a positive or negative impact on their work.

The six aspects identified in the model as being affected by received feedback (Figure 8.2) are the result of a combination of literature and findings. During the Feedback in One Software Engineering Environment Study (chapter 5) software engineers reported that job satisfaction, productivity, motivation, enjoyment and code quality could be impacted by received feedback. In this model, productivity and code quality are combined and represented by the aspect performance. Enjoyment is included in the aspect feelings. These findings informed the model by identifying both the instant and delayed impact of received feedback on software engineers, and the aspects which can be affected by received feedback.

ONLINE SURVEY

In The Effect of 'Source' and 'Medium' Feedback Characteristics study (chapter 7) software engineers were asked to respond to twelve scenarios or questions investigating feedback in software development environments. This study influenced the model by finding: (1) software engineers have preferences for the medium used to receive feedback, (2) feedback is more or less valuable depending on the source of the feedback, (3) feedback that is reported as the most valuable may not have the most impact, and (4) software engineers indicated that all six aspects can be affected by received feedback.

1. **Medium.** Three scenarios investigated if the respondents had preferences to receiving feedback through certain media, and if receiving feedback through a non-preferred medium would change the impact of the feedback. Over 50% of the respondents indicated multiple preferences, and over 50% of the respondents indicated that receiving feedback from a non-preference would change the impact of the received feedback. These findings suggest that the medium through which feedback is received is a preference for software engineers, and that receiving feedback through a non-preferred medium can change the impact of the received feedback.
2. **Source.** Building on previous findings, nine of the twelve scenarios or questions in this study investigated the impact of the feedback source. The respondents indicated differing values for the six different sources (new software engineer, experienced software engineering, senior software engineer, line manager, senior manager, company CEO) of feedback. Over 50% of the participants indicated that feedback from a senior or experienced software engineer was more valuable to them than feedback

from a senior manager, and over 80% indicated that feedback from a senior or experienced software engineer was equal to or more valuable than feedback from their line manager. In all questions, the least valuable was feedback from a new software engineer. These findings indicate that the source can affect the impact of received feedback, specifically by the type of source (colleague or manager) and the source’s experience (new, experienced, or senior).

3. **Values vs. Impact.** During the online survey the respondents were asked to indicate how valuable they believed feedback to be from six different sources (new software engineer, experienced software engineering, senior software engineer, line manager, senior manager, and company CEO). Later, they were asked to indicate the impact received feedback from three sources (software engineer, line manager, and senior manager) would have on six different aspects (feelings, motivation, behaviour, job satisfaction, attitude, and performance). Sub-groups of respondents were identified from their reported value of feedback from the different sources. An analysis of these sub-groups identified that the feedback reported as being the most valuable, was not the same as feedback reported as impacting the most aspects. This finding suggests that the feedback software engineers report as the most valuable is not the same as the feedback which has the most impact. The model includes this finding by identify the hygiene assessment that can affect the impact of received feedback.
4. **Aspects.** When analysing the responses to the negative feedback scenarios, over 80% of the respondents reported that they would want to “fix the problem”. Attitude was reported by a much smaller percentage of respondents as being affected by received feedback, but as it was reported as being affected by received feedback it still warrants consideration. Job satisfaction, feelings, motivation and performance were all reported as being affected by received feedback. The model includes this finding by identifying behaviour, job satisfaction, feelings, motivation, performance, and attitude as aspects that can be affected by received feedback.

LITERATURE

As discussed earlier (section 8.1.1), individual perceptions may be altered by their current state of mind, their mood or their emotions. The model addresses this by including an individual's affective state as a section in the model that can influence the individual's perceptions.

The literature review (chapter 2) identified studies reporting that motivation could affect an individual's productivity (Procaccino *et al.*, 2005) and the quality of their software (Boehm, 1981). In the model, these are combined and represented by the aspect performance. Job characteristics theory, (Hackman and Oldham, 1976) identified feedback as a factor affecting motivation, in turn affecting an individual's satisfaction with their work and the quality of their work. In the model, job satisfaction is included directly, and the quality of their work is represented by the aspect performance.

MODEL FOUNDATION SUMMARY

The previous sections identified and discussed the empirical data and literature that influenced the model of feedback in software engineering. The model combines data collected from a range of different participants, collected using different research methods, and investigating feedback with several different focuses as the research progressed. Overall, the model represents how feedback is perceived, evaluated, and reacted to by software engineers in software engineering environments. Table 8.1 summarises the data presented in this section and how it contributes to the model of feedback in software engineering.

TABLE 8.1 – RELEVANT DATA AND LITERATURE INFLUENCING THE MODEL OF FEEDBACK IN SOFTWARE ENGINEERING

Model Section	Relevant Data / Literature	Chapter
Feedback - **Feedback characteristics**	Preliminary study Literature identifying feedback characteristics in clinical education	Two and Three
Individual Perceptions - **Feedback characteristics**	Preliminary study	Three
Current state of mind / mood / emotions	Literature investigating the impact of an individual's affective state	Eight
Individual *value set* in Software Engineering - **Source Value** - **Feedback Value**	Personal profiles	Five
Individual *value set* in Software Engineering - **Feedback Preferences (Setting)**	Personal profiles	Five
Individual *value set* in Software Engineering - **Feedback Preferences (Medium)**	Scenario Responses Online Survey	Six and Seven
Assessments - **Source Validity** - **Source Value** - **Feedback Validity** - **Feedback Value** - **Medium Preference** - **Setting Preference**	Scenario Responses Online Survey	Six and Seven
Assessments - **Hygiene**	Online Survey	Seven
Impact - **Instant** - **Positive** - **Negative** - **Feelings**	Diary Feedback Instances	Five
Impact - **Delayed** - **Positive** - **Negative** - **Job Satisfaction, Feelings, Motivation, Performance**	Diary Day Summaries	Five
Impact - **Job Satisfaction, Motivation, Performance**	Preliminary study Literature investigating the impact of feedback	Two
Impact - **Feelings, Motivation, Job**	Scenario Responses Online Survey	Six and Seven

8.3 EXAMPLE MODEL PROCESSES

This section presents instances of feedback collected during the Feedback in One Software Engineering Environment Study (chapter 5) , and compares them to the model of feedback shown in Figure 8.1 and Figure 8.2. The instances of feedback used were all collected from data presented in chapter 5 from the diary study. However, data from other stages is included in the 'Individual *value set*', which uses all the data collected about each individual to identify the individual *value set* specific to software engineering, as shown in the personal profiles discussed earlier.

The instances of feedback presented in this chapter represent the range of different forms of feedback reported by software engineers. The instances of feedback collected in the diary study (chapter 5) contained a mixture of selected single-phrase responses and free-text responses. The instances of feedback included in this chapter were selected because they both represented the range of different forms of feedback, and included free-text responses. Instances of feedback that included free-text responses often provided useful information aiding the understanding of the received feedback and the effect it had on the feedback recipient.

Individual instances often do not articulate every aspect of the model. Where an aspect of the model is interpreted from the data but not reported explicitly, it is italicised. The characteristics listed are only those identified from an analysis of each participant's data specific to software engineering, and are not a complete representation of each participant's individual *value set*. To aid interpretation, keywords included in the model (Figure 8.2) are emboldened, and data that was interpreted during the diary study or during in another research method is italicised. It should be noted that the feedback instances did not investigate the impact of feedback on the six aspects identified in Figure 8.2, but rather focused on how feedback impacted the recipient's feelings.

Feedback Instance #59

Individual Perceptions:	**The Source:**	Manager
	The Content:	Positive, Performance and Attitude, Opinions on direction future work should take
	Other Feedback Characteristics:	Face to face, Meeting
Individual *value set*:	*Preferences concerning the medium used, do not focus on feedback from people above me in the company hierarchy, value feedback from people with good technical ability and/or technical knowledge, focus on the credibility of the source.*	
Assessments:	*Valid and valuable feedback, with an acceptable goal in line with my preferences.*	
Impact:	Positive impact on feelings – "Happy, Reassured, Energised". *Positive impact on feelings* – "This feedback came as part of my formal 1:1 with my manager. The manager had canvassed opinion from my colleagues and so I was able to hear that there are very happy about my contribution to the team. This was really important to me as new starter here in the company, as it is important to me to know that I am in role which I can fulfil well."	

In feedback instance 59 the participant reported receiving positive feedback from his manager, however the feedback is actually more about what his colleagues are saying. As the manager is relaying feedback to the recipient from the recipient's colleagues, the feedback is valid and valuable, especially as he is new within the company, and it confirms that he is progressing well and fitting in his team and with his new colleagues. The effect of the feedback appears to be more dependent on what the manager is relaying than the manager's own direct feedback, and it is possible that if this feedback did not include the relayed colleague feedback component, the impact would be different.

Feedback Instance #45

Individual Perceptions:	**The Source:**	Senior Manager
	The Content:	Positive, Performance
	Other Feedback Characteristics:	Email, Casual

Individual *value set*:	*Preferences as to the medium used, no focus on feedback from people above me in the company hierarchy, value feedback from people with good technical ability and/or technical knowledge, focus on the credibility of the source.*
Assessments:	*Not valid nor valuable feedback, from a source that is not valid, with an unacceptable goal.*
Impact:	Negative impact on feelings – "Annoyed, Unhappy" *Negative impact on feelings* – "Her feedback seemed obviously fake, and only there to get us to be happy to what she wanted us to do"

In feedback instance 45 the feedback received is positive, but the impact is negative. The participant's characteristics suggest that he values feedback from sources with technical knowledge and ability, and not from people above in the hierarchy. This feedback came from a senior manager, and the perceived goal of the feedback was unacceptable, as it was intended to get the recipient to do "what she wanted us to do". Without the unacceptable goal, it is possible the received feedback may have a less negative impact, but with an unacceptable goal the feedbacks impact is negative.

Feedback Instance #42

Individual Perceptions:	**The Source:**	Project Manager
	The Content:	Positive, Performance
	Other Feedback Characteristics:	Email, Casual, "'you're awesome' – I'm not even making this up!"
Individual *value set*:	*No focus on feedback from people with good technical ability and/or technical knowledge, value feedback from people above me in the company hierarchy.*	
Assessments:	*Valid and valuable feedback, with an acceptable goal in line with my preferences.*	
Impact:	Positive impact on feelings – "Happy, Reassured, Inspired". *Positive impact on feelings* – "I'm having to argue a bit with folks from other teams and divisions, and they don't perhaps have the same urgency or motivation to assist us with our current goals. This sort of feedback gives me more confidence when dealing with them and being assertive, which is something I perhaps need to improve at. Pow! Take *that*, other people!"	

In feedback instance 42, the participant received feedback from his project manager telling him that he is awesome. This feedback is from his project manager, so the source is valid and valued, and the feedback content is considered valid and valued as it is reassuring him of his ability and helping him to be more assertive. The participant's "Pow! Take *that*" comment indicates that feedback increases his confidence and may affect his behaviour in the future as he reports that receiving this feedback "gives [him] more confidence when dealing with them".

Feedback Instance #21

Individual Perceptions:	**The Source:**	Visual Studio, the Web
	The Content:	Negative, Performance
	Other Feedback Characteristics:	Debugging output + websites, Casual, "Discovered I'd missed something reasonably significant in terms of how ASP.Net WebServices function, and we're launching a product that depends on a WebService I've written tomorrow - oh no!"
Individual *value set*:	*No focus on feedback from people with good technical ability and/or technical knowledge, value feedback from people above me in the company hierarchy.*	
Assessments:	*Valid and valuable feedback, with an acceptable goal in line with my preferences.*	
Impact:	Negative impact on feelings – "Concerned, Unhappy, Anxious" *Positive impact on feelings* – "Inspired, Excited" *Negative impact on feelings* – "Made me feel a bit stressed whilst I did a scout for what implications this would have for my product. It hadn't been caught because I'm new to this stuff and it hadn't actually manifested in the form of a bug, but I certainly felt pretty stupid / sheepish because of it. Subsequently discovered it required only minor change, resulting in a sense of relief, but still angry that I hadn't noticed this before now. Worried that there's maybe more that I've missed? Spoken to people about it, they thought it was amusing / not a big deal, so I now feel more calm."	

In feedback instance 21, the participant received feedback from the output of the debugging system, and when he investigates he realises this is negative feedback. The feedback is valid and valued, and subsequently has an instant negative impact on his feelings. However, later when he realises the issue is not as bad, his perception of the feedback changes, and he reports the delayed impact of feeling calm, after being excited and inspired to fix the issue.

Feedback Instance #29

Individual Perceptions:	**The Source:**	Colleague, Project Manager
	The Content:	Positive, Performance
	Other Feedback Characteristics:	Face to face, Casual, "So they wanted me to demo HRT to the other sales guys while they're here. That's pretty neat! I'm glad it's worth showing to people."
Individual *value set*:	*No focus on feedback from people with good technical ability and/or technical knowledge, value feedback from people above me in the company hierarchy.*	
Assessments:	*Valid and valuable feedback, with an acceptable goal in line with my preferences.*	
Impact:	Positive impact on feelings – "Happy, Excited". *Negative impact on feelings* – "Anxious"	

In feedback instance 29, the participant received feedback from his project manager and colleague asking him to demonstrate a piece of software to some visitors. This request was considered feedback, as his software is worthy of showing to people. This feedback is valid and valuable, as is the source, so the impact is positive.

Feedback Instance #50

Individual Perceptions:	**The Source:**	Project Manager, Colleague
	The Content:	Positive and Negative, Performance, Attitude, Behaviour,
	Other Feedback Characteristics:	Face to Face, Meeting and Review/Report "Positive feedback on internal job interview, resulting in new role. Very encouraging, very pleased ;-D Quite a long ad-hoc meeting, not originally scheduled in calendar. Some slightly negative feedback in terms of things to watch out for, but delivered well and constructively, which is necessary."
Individual *value set*:	*No focus on feedback from people with good technical ability and/or technical knowledge, value feedback from people above me in the company hierarchy.*	

Assessments:	*Valid and valuable feedback, with an acceptable goal in line with my preferences.*
Impact:	Positive impact on feelings – "Happy, Energised, Excited, Pleased, Proud, Inspired" *Positive impact on feelings* – "Definite spring in my step! Wrote down the negative feedback on a pink post-it note : it's becoming a tradition to capture anything like this, then park it so that I can carry on with the rest of my day - writing down means it gets remembered, but moving out the way means I can maintain focus."

In feedback instance 50, the participant reported receiving feedback on his successful application for a new position within the company. The feedback was from his project manager, and contained both positive and negative comments – however the impact is purely positive, causing the participant to feel six different feelings

The instances of feedback presented in this sub-section show that the impact of received feedback can be different, and that this change can be understood by comparing the instances of feedback shown in this section to the model of feedback in software engineering (Figure 8.1 and Figure 8.2).

8.4 SUMMARY

In summary, this chapter presented a model of feedback in software engineering, giving both an overall and a more detailed view of how software engineers receive feedback by combining relevant literature and the findings from this research. Software engineers perceive the characteristics of the received feedback, and use this information to provide knowledge, which combined with the individuals' characteristics is used to influence several assessments, which then results in the impact of the received feedback.

The process of perception-knowledge-assessment-impact is a recurring cycle, as perceptions are made at different points during and after receiving feedback. This cycle indicates that the initial impact of received feedback may not be the same as the delayed impact, as software engineers reflect and reconsider the feedback they've received.

The model provides an empirically-grounded representation of how feedback is received in software engineering environments, identifying the core stages involved when software engineers receive feedback. Future work should look to extend the model, clarifying the assessment making process, investigating other individual *value set* factors, and addressing the influence of aspects external to the model discussed in the ‘current state of mind / emotions’ to ensure that all relevant stages and affects are considered within the model.

CHAPTER 9

DISCUSSION

This chapter discusses the key findings from four studies with software engineers, addressing and answering the overall research question: *How does feedback impact the motivation of software engineers?* Included in this chapter is a section that discusses the findings in the context of theories of motivation, identifying differences in how feedback is defined and discussed by software engineers, and how feedback is defined and discussed by theories of motivation.

The following sections discuss the definition of feedback and how feedback was identified by software engineers (9.1), and discusses the importance of feedback characteristics (9.2). The chapter continues to discuss the difference between feedback value and feedback impact as reported by software engineers (9.3), uses the findings of this study to compare to theories of motivation (9.4), discusses recommendations for feedback in software engineering environments (9.5), and finally presents the limitations of the findings from this research (9.6).

9.1 THE DEFINITION OF FEEDBACK IN SOFTWARE ENGINEERING

The literature review (chapter 2) identified that in multiple disciplines feedback is defined as a single construct. Wiener (1950: 58-59), a psychotherapist, described the process of providing feedback as “telling [effective behaviour] whether it has equalled its goal or fallen short”. A definition of feedback is “information about reactions to a product, a person’s performance of a task, etc. which is used as a basis for improvement” (dictionary.com, 2012), and the Job Characteristics Theory of motivation defines feedback as “the individual obtaining direct and clear information about the effectiveness of his or her performance.” (Hackman and Oldham, 1967: 258) Research has challenged the definition of feedback as a single construct; an investigation in clinical education identified nine characteristics of feedback that together define feedback (van de Ridder *et al.*, 2008).

Various studies have identified and investigated what can be described as feedback characteristics. Shanab *et al.* (1981) investigated the effects of positive and negative verbal feedback, and how it affected the intrinsic motivation of participants; Herold and Greller (1975) “attempted to dimensionalize feedback” (Herold and Greller, 1977: 142) by investigating the effect of five different sources of feedback. Both the Herold and Greller and the Shanab *et al.* studies identified a single feedback characteristic and investigated the effect of the different values for the identified feedback characteristic.

The findings of the Feedback Scoping Study (chapter 3) support the findings of van de Ridder *et al.* and suggest that feedback is defined by software engineers as a combination of ten feedback characteristics. The identification of feedback characteristics comparable to those reported in clinical education (van de Ridder *et al.*, 2008) underpinned all subsequent empirical studies completed during this research.

The identification of ten characteristics of feedback in software engineering suggests that previous studies investigating feedback in software engineering may have not adequately considered three features of feedback: (1) the definition of feedback, (2) the identification of feedback characteristics, and (3) the effect of different forms of feedback. Future research investigating feedback in software engineering should consider these aspects.

9.2 THE IMPORTANCE OF FEEDBACK CHARACTERISTICS

The findings of the empirical studies completed during this research (chapters 3, 5, 6 and 7) highlight the importance of the feedback characteristics. Feedback was found to affect six aspects, and the impact of received feedback varied. The variation in the effect of received feedback was identified as the result of the feedback recipients’ perceptions of the characteristics of received feedback, influenced by the feedback recipients’ individual *value set* (chapter 8). Identifying the characteristics of feedback, and evaluating the effect of the different values of feedback characteristics, has established the importance of feedback characteristics in defining feedback and understanding the effect feedback has on software engineers.

The characteristics of feedback may have been overlooked by most of the literature, irrespective of discipline. Herold and Greller (1977: 146) reported that “much of the work on job attributes fails to distinguish feedback by its valence”, referencing work by Hackman & Lawler, 1971; Hackman & Oldham, 1975; Jenkins *et al.*, 1975. Hackman and Lawler (1971: 282) stated, when discussing feedback, that “the exact meaning of the dimension remains highly uncertain." Ramaprasad (1983: 4) discusses feedback in management theory, and argues that “despite wide usage of the concept of feedback, there is little consensus among management theorists on the definition of the concept.” All of the listed references are from the 1970s and 1980s, however the problem still exists today; Hattie and Timperly (2007: 81) investigated feedback in education and reported that “few recent studies have systematically investigated the meaning of feedback [in education]”.

Research that investigates feedback either references the definition of feedback used by one theory of motivation, or appears to accept that feedback is a commonly understood and defined concept that does not require a clearly-stated definition. Previous research that included feedback in studies of motivation in software engineering often investigated the applicability of a theory of motivation in software engineering, or referenced a theory of motivation. Subsequently, the definition of feedback in studies investigating motivation in software engineering is often the definition of feedback in one theory of motivation referenced by the research. The Job Characteristics Theory (JCT) (Hackman and Oldham, 1976) defines feedback as:

> *“The degree to which carrying out the work activities required by the job results in the individual obtaining direct and clear information about the effectiveness of his or her performance.” (ibid: 258)*

Cheney (1984) investigated programmer productivity and used the JCT definition of feedback; LeDuc (1980) investigated the motivation of programmers, also using the JCT definition of feedback; Couger *et al.* (1989) compared the environments of programmers in the US, Israel and Singapore using a survey based on the JCT, subsequently using the JCT definition of feedback; Couger and Ishikawa (1985) compared the environments of programmers in the US and Japan using a survey based on the JCT, subsequently using the JCT definition of feedback; Gambill *et al.* (2000) investigated a holistic model of task design for IS personnel and included both JCT and Goal Setting Theory (Locke, 1968), including the definition of feedback from both theories of motivation.

However, previous research has included feedback without defining feedback. Wegge and Haslam (2005) investigated how they could improve motivation and performance in brainstorming groups, and identified the use of feedback to improve performance by making tasks more valuable and intrinsically motivating, however no reference to or statement of the definition of feedback is presented; Stenmark (2000) encourages one form of feedback (work-focused) as opposed to another (person-focused) when managing creative work, but does not discuss the definition of feedback; Voas (2001) reflects on managing a software "superstar" and advocates providing both positive and negative feedback, but does not define feedback. Nor is this a changing trend; Linderbaum and Levy (2010) investigated a feedback orientation scale, a scale intended to address individual differences when providing performance feedback. Linderbaum and Levy (2010) reference studies identifying the effect of feedback and discuss the studies in detail, but provide no definition of feedback and do not identify any characteristics of feedback.

Software engineers, across three empirical studies (chapters 5, 6 and 7), consistently reported a change in the impact of received feedback attributed to the values of the characteristics of the received feedback. The source, the medium, and the polarity of the feedback content were identified during the three studies (chapters 5, 6 and 7) as characteristics whose values influenced the effect of received feedback. The findings of the earlier studies (chapters 3 and 5) influenced the later studies (chapters 6 and 7) to focus on the effect of the medium and source feedback characteristics; the findings of the later studies (chapters 6 and 7) suggested that software engineers have a preference for the medium used to receive feedback, and that the source of the feedback influences the effect of the received feedback due to an assessment of the sources value and validity (chapter 8) by the feedback recipient. The findings of three studies (chapters 5, 6 and 7) highlight the need to consider the values of feedback characteristics when investigating the impact of feedback in software development environments, and identified the effect that the values of received feedback can have on software engineers.

9.3 SOFTWARE ENGINEER PERCEPTIONS OF FEEDBACK VALUE AND IMPACT

One of the questions in the online survey (chapter 7) asked the engineers to rate how valuable to them feedback was from six different possible sources of feedback (new software engineer, experienced software engineer, senior software engineer, line manager, senior manager, company CEO). The results of the online survey (chapter 7) indicated that software engineers valued feedback from an experienced or senior software engineer the most, but also indicated that feedback from line or senior manager would have more effect on the participants. This finding was attributed to software engineers understanding work-hygiene implications, the potential impact of being seen to be doing good and/or bad work by people who can influence their job at the company, their future project role, and their progression within the organisation.

Hygiene Theory (Herzberg, 1959) identifies factors which are believed to be motivating and provide job satisfaction, and factors which are believed to avoid job dissatisfaction, named hygiene factors. Hygiene Theory lists ‘pay’, ‘status’, ‘interpersonal relations with superior’, ‘work conditions’, and ‘job security’ among ten hygiene factors. A line manager or senior manager could directly influence the job security, work conditions, and pay of a software engineer. The effect that receiving feedback from a line manager or senior manager may have on the identified hygiene factors may explain why feedback from line managers and senior managers is reported as affecting the most aspects.

The findings presented in this section are attributed to a work-hygiene concern for software engineers found in a theory of motivation. An alternative account is that software engineers do not link value with impact. Future research should investigate this finding, and elicit from software engineers the reason for the difference in feedback value and feedback impact.

9.4 FEEDBACK IN THEORIES OF MOTIVATION

Four theories of motivation that include feedback as a factor are: Hackman and Oldham’s Job Characteristics Theory, Herzberg’s Hygiene Theory, Locke’s Goal Setting Theory, and McClelland’s Theory of Needs. In the following paragraphs each of the four theories of motivation are discussed in relation to the findings from this research.

JOB CHARACTERISTICS THEORY

Job Characteristics Theory (JCT) identifies five factors required for a job to be motivating, including feedback. Feedback is defined in JCT as "the degree to which carrying out the work activities required by the job results in the individual obtaining direct and clear information about the effectiveness of his or her performance." (Hackman and Oldham, 1976: 258) Hence feedback is something that provides an individual with information about his or her performance. Different sources of feedback are discussed by Hackman and Oldham (1976). JCT only accounts for feedback "derived from the job itself" (ibid: 272), but it is acknowledged that "feedback is received by employees from many additional sources: supervisors, peers, and so on" (ibid: 272) and that "there is reason to believe that feedback from various sources may interact with one another in affecting individuals' knowledge of the results of their work and their affective reactions to the job as a whole" (ibid: 272). However, the effect of the source of feedback has not been established.

The description of feedback in JCT does not address the different characteristics of feedback and the effect that the values of feedback characteristics may have on motivation. Feedback is presented as an instrument to provide an individual with the knowledge of the outcome of their performance. By providing an individual with 'knowledge of the actual results of work activities' which, when combined with two other experienced psychological states (experienced meaningfulness of the work, experienced responsibility for outcomes of the work) leads to four personal and work outcomes (high internal work motivation, high quality work performance, high satisfaction with the work, low absenteeism and turnover).

The sources of feedback identified in the empirical studies (chapters 3, 5, 6 and 7) support and extend the findings Couger and Zawacki (1980) by identifying feedback from the supervisors as well as users and senior managers, but also identifying that the most common form of feedback received by software engineers was not included by either the JCT model or the Couger and Zawacki (1980) model: feedback from peers. During the studies in this research, software engineers reported the importance of their fellow software engineers (chapter 5), that the most common form of feedback was from software engineers (chapter 5), and that the most valued feedback was received from software engineers (chapter 7).

The JCT model does not consider the effect of different forms of feedback, does not identify the different characteristics of feedback, and does not address the impact that feedback can have outside of providing an individual with the knowledge of the outcome of their performance. This research identified feedback as a combination of ten characteristics (chapter 3), observed that the values of the feedback characteristics can affect the impact of received feedback (chapters 5, 6, and 7), and established that received feedback can affect software engineer's behaviour, attitude, performance, motivation, job satisfaction, and feelings. The comparisons of the JCT model and the findings from this research suggest that the JCT model does not consider the effect of feedback sufficiently for use in software engineering environments.

HYGIENE THEORY

Hygiene Theory proposes that there is a difference between factors which provide satisfaction (motivators), and factors which avoid dissatisfaction (hygiene factors). Hygiene Theory identifies the two strongest factors associated with job satisfaction as achievement and recognition. Hygiene Theory does not explicitly name feedback as a motivator or hygiene factor, however recognition is reported as typically occurring through "verbal recognition" (Herzberg, 1980: 45), with examples including praise, blame, and criticism, which are all forms of feedback. The findings of this research support that recognition can be received through feedback. Software engineers reported during the diary study (chapter 5) that received feedback made the individual feel like "your work at the company is appreciated"; that "personal recognition is what is most satisfying" upon receiving feedback; and that receiving feedback would help the individual know that "people are appreciating the work" of the individual.

In Hygiene Theory recognition is not distinguished by the value or valance it has to the recipient, and the source of the recognition, the value or valance of the source to the recipient, and any other characteristics of feedback are not identified or discussed. Hygiene theory states that "the source could be almost anyone: supervisor, some other individual in management, management as an impersonal force, a client, a peer, a professional colleague, or the general public", however hygiene theory does not then identify the effect of a different source of recognition and how it impacts the recipient's motivation.

GOAL SETTING THEORY

Goal Setting Theory proposes that people are motivated to achieve the successful completion of challenging goals, claiming that more difficult goals result in higher performance than easy goals, specific goals produce higher performance than general goals, and that the behavioural intentions of each individual influence the choices that they make. Goal Setting Theory identifies three forms of feedback, 'Knowledge of score' and 'praise and/or reproof', as affecting an individual's motivation. Knowledge of score is providing information about the individual's performance, informing them of how well they performed. Knowledge of score can be given in relation to a standard, for example a green light used to signify that an individual had met a required reaction time on a task (Locke, 1968). Praise and/or reproof are two different forms of feedback, where positive or negative feedback is given to an individual relating to their performance.

Goal Setting Theory proposes that providing individuals with feedback on the outcome of their actions through the 'knowledge of score' may suggest a goal for the individual, and if the knowledge of score is in relation to some external standard, it is "certain to imply a goal" (ibid: 184) for the individual. 'Praise and reproof' are reported as indirect goal-setting incentives, and that each individual's reactions to praise and reproof will depend on multiple factors: "whether he considers the comments just or unjust, the particular work context in which the comments were made, his liking and respect for the person making the comments, his own personality" (ibid: 185). These factors that can change the impact of praise and/or reproof are related to characteristics of feedback. Respect for the source, perceived validity of the comments and the source's ability to provide them, and contextual factors are all related to feedback characteristics identified in previous literature and in this research (chapter 3).

Goal Setting Theory identifies the validity of praise and reproof comments by arguing that an individual will consider comments "just or unjust" (ibid: 185). The findings of this research (chapter 8) identified a different set of considerations by the individual: validity and value of the source, validity and value of the content. The content or source could be considered valid, but if the feedback were of no value the feedback might not have an affect (chapter 8). Similarly, the content or source could be of value, but if considered invalid the feedback may not have an effect (chapter 8). During the diary study (chapter 5), interviews (chapter 5), and the scenario study (chapter 6) software engineers reported ignoring valid feedback because the feedback was received too frequently, or it was perceived as the source's 'job' to provide the individual with feedback. While opinions ranged, some engineers reported during the scenario study (chapter 6) that receiving feedback from a potentially less valid source, such as a division manager who had less knowledge of the precise details of the engineers work, would be more valuable for the engineer as it occurred less frequently.

Goal Setting Theory identified the feedback recipients' opinion of the source of the received feedback as "respect for the person making the comments" (ibid: 185). Findings from the empirical studies (chapters 5, 6, and 7) suggest that respect for the source can be separated into two categories: respect for the technical knowledge of the source, and respect for the hierarchical position of the source. Technical respect for a source was reported during interviews (chapter 5) and reinforced during a scenario study (chapter 6) as the feedback recipient's perceptions of the source's knowledge and understanding of programming and programming principles. Some engineers valued technical respect more than others, affecting the impact of received feedback (chapters 6 and 7). The impact of the hierarchical position of the source was seen in multiple studies (chapters 5, 6 and 7) in which software engineers provided different reports on the effect of the source's hierarchical position.

Goal Setting Theory identifies the effect of three forms of feedback and discusses different feedback characteristics, but does not address the effect of the values of the feedback characteristics, how the values affect the feedback, and how this fits in the Goal Setting Theory. The findings of this research suggest that the validity of feedback in Goal Setting Theory can be extended in software engineering by including source value and validity, and content value and validity. For suitability in software engineering environments, Goal Setting Theory could be extended to consider the impact of the source, relating to the individual *value set* (chapter 8) of the feedback recipient, and how the source affects the impact of feedback.

THEORY OF NEEDS

Theory of Needs argues that each individual has a need for achievement, a need for power, and a need for affiliation. McClelland (1961) characterised an activity as achievement-oriented when the individual feels responsible for the outcome, the individual expects to receive unambiguous feedback on the results of his or her actions, and when there is a degree of risk or uncertainty. Upon completing an activity and receiving unambiguous feedback on the outcome, each individual should experience a sense of pride in their achievement. Similar to Goal Setting Theory, Theory of Needs proposes that achievement-oriented activities which are less certain and harder but not impossible to complete, will result in a stronger sense of achievement than an activity which was easy and completion was a foregone conclusion.

Theory of Needs does not identify any characteristics of feedback, nor how the values of characteristics of feedback may affect received feedback. Unlike other theories of motivation, Theory of Needs does not consider feedback validity or how the recipients' perception of validity may change the effect of feedback. The studies in this research (chapters 3, 5, 6 and 7) did not directly investigate feedback in relation to achievement-oriented activities, but the findings from this research indicate that the source and their perceived validity to provide feedback can affect the impact of received feedback. This suggests that experiencing a sense of pride upon receiving feedback in achievement-oriented activities may be affected by the source of the feedback, if the source is not perceived as valid.

Need for affiliation is identified as the need to create and maintain relationships with others. McClelland (1961) reports that people with high need for affiliation will work hard to be accepted by others, and may even display a high need for achievement if they believe it will increase their chances of acceptance by others. Findings from interviews with software engineers (chapter 5) support the need for affiliation, as engineers reported the importance of being seen to be doing good work by fellow software engineers as well as line and senior managers. Technical respect for the source of feedback was identified as a factor for software engineers in several studies (chapters 5, 6 and 7). Technical respect may affect the engineers need for affiliation, and who they wish to increase their affiliation with. Findings from an online survey (chapter 7) indicated that software engineers valued feedback from new software engineers less than feedback from experienced or senior software engineers, suggesting that the engineers need for affiliation might be higher with experienced or senior software engineers compared to new software engineers.

Theory of Needs explicitly identified feedback as an integral part of the three factors defining an achievement-oriented activity, and identifies unambiguous feedback as a method of providing the individual with the results of his or her actions. Theory of Needs does not consider the impact of the source, the medium, or the polarity of the feedback, all identified to affect the impact of received feedback during this research (chapters 5, 6 and 7). To be suitable for software engineering environments, Theory of Needs could be extended to consider the impact of the value of feedback characteristics, and how an individual's sense of accomplishment and their need for affiliation can be affected by the values of feedback characteristics.

MOTIVATION THEORY SUMMARY

These four theories of motivation all address feedback in terms of providing an individual with the results of his or her actions or with recognition for something the individual has done. Characteristics of feedback can be found in the four theories of motivation, but the impact of a feedback characteristic is not discussed, nor are they explicitly identified as feedback characteristics. Some of the characteristics of feedback as identified by this research (chapter 3) and research in clinical education (van de Ridder *et al.*, 2008) are included in theories of motivation and are identified as factors that can affect the impact of feedback, but the effect of the discussed characteristic is not investigated or identified by any of the theories.

This research suggests that in software engineering, feedback provides more than just the knowledge of the results of an individual's actions or recognition. The values of feedback characteristics were found to affect the impact of received feedback (chapters 5, 6 and 7), and software engineers reported that six different aspects, including motivation, were affect by received feedback (chapters 5, 6 and 7). In software engineering environments, the impact of received feedback is more than just knowledge of results, and theories of motivation when used in software engineering may need to be adapted to consider the findings of this research.

9.5 RECOMMENDATIONS FOR SENDING FEEDBACK IN SOFTWARE ENGINEERING ENVIRONMENTS

The findings of this research suggest a few recommendations that can be made to enhance the feedback sender-receiver relationship, improve the desirability and effect of received feedback, and reduce the undesired effect of received feedback in software engineering environments. During the diary study (chapter 5) some instances of feedback were perceived as positive, but had a negative impact; some instances of feedback were perceived as negative, but had no effect; some instances of feedback were considered invalid and ignored. These instances appear to have an effect that is not desired by the feedback source, and in the case of a negative impact, is probably not desired by the feedback recipient. The recommendations set out below aim to address some of the causes behind these instances of effective feedback, with the intention of creating a mutually beneficial sender-receiver feedback relationship.

- **Intention-impact disparity**

Anyone providing or receiving feedback in software engineering environments should consider that the intention of the feedback, and the effect of the received feedback, may not be the same. Empirical findings suggest that software engineers consider the values of the characteristics of received feedback (chapters 6 and 7), and make assessments about the feedback resulting in the impact of the received feedback (chapter 8). As seen in a diary study (chapter 5), one software engineer reported a negative effect on their feelings (annoyed, unhappy) because a received email was perceived as “obviously fake” and tried to get the recipient to do “what she wanted”.

- **Individual preferences**

Software engineers reported a preference for the medium and setting of received feedback (chapters 6 and 7) and discussed that the medium used to send the feedback may suggest something about the importance of the feedback (chapter 5). However the ‘importance’ of the feedback depending upon the medium used varied; some engineers reported receiving feedback via email would suggest it was more important, and some engineers reported receiving feedback via email would suggest it was less important (chapter 5). Attempts should be made to understand the medium preference and setting preference of the feedback recipient, as 50% of software engineers reported that feedback received through a medium that was not preferred would have a minor (20%), moderate (20%) or major (10%) effect on the impact of the received feedback.

- **Feedback content**

When providing feedback to software engineers, the recipients' individual *value set* should be considered. Software engineers reported different values for different forms of feedback (chapters 5 and 6). Understanding what feedback is of value to the recipient will ensure that any provided feedback will not be ignored due to the feedback content.

- **Perceptions of the source**

Anyone providing feedback to software engineers should consider how their knowledge, experience, ability, previous working history, and hierarchical position will be perceived by the feedback recipient. Software engineers reported that that feedback received from a source perceived as unable to comment validly would be ignored (chapter 5), and the importance of the technical knowledge of the feedback source when providing technical feedback (chapters 5 and 6). The recipients' assessment of the validity of received feedback was associated with their perceptions of the feedback source, and if the recipient perceived the feedback source as a both valuable and valid source of feedback. Sometimes, asking someone else to pass on the feedback can be more effective.

- **Source relationship**

Maintaining a solid relationship between the feedback source and feedback recipient may address many of the negative or undesired responses to received feedback. Gaining an understanding of the feedback recipients' preferences to the medium and setting of received feedback, their perceptions of the feedback sources ability and knowledge, and their individual *value set* relating to feedback value and feedback validity increases the probability of aligning the intention and the impact of the received feedback.

- **The impact of received feedback is not just knowledge of results**

Job satisfaction, motivation, performance, attitude, behaviour and feelings were all reported by software engineers (chapters 5, 6 and 7) as being affected by received feedback. When providing software engineers with feedback, the impact that the feedback will have in addition to providing the recipient with the knowledge of the outcome of their actions, should be considered.

- **Feelings are important**

Research has identified that an individual’s affective state can affect their perceptions and their decisions (chapter 8). Amabile and Kramer (2011: 52) report that “creativity follows from positive emotion” and that “the more positive a person’s mood on a given day, the more creative thinking he did the *next* day-and, to some extent, the day after that” (ibid: 52). Software engineers reported that, typically for every one instance of received feedback, two feelings were experienced (chapter 5). Providing feedback that is likely to affect the recipients’ feelings will have an effect on how the feedback is perceived and reacted to. Anyone providing feedback to software engineers should consider the potential emotional effect of the feedback, and how it may distort the underlying message or intention of the feedback.

If these recommendations are considered by anyone providing software engineers with feedback, the undesirable outcomes of received feedback may be avoided, reducing the negative effect of received feedback while ensuring that the desired message of the feedback is effectively communicated.

TABLE 9.1 – CONSIDERATIONS WHEN PROVIDING FEEDBACK

How well do you know the feedback recipient:

Which medium does the recipient prefer?

Which setting does the recipient prefer?

Does this feedback have value to the recipient?

Are you perceived by the recipient as a valuable source of feedback?

If you are providing feedback that requires knowledge of the situation:

Are you perceived by the recipient as having sufficient knowledge of the situation?

If you are providing technical feedback:

Are you perceived by the recipient as being technically-able?

If you are consistently providing feedback to the recipient:

Has your feedback become repetitive, boring, or expected?

The impact of feedback:

Will this feedback have the impact that you intend it to have?

What effect will this feedback have on the recipient's attitude, behaviour, job satisfaction, performance, motivation, or feelings?

If the feedback is negative, what short-term and long-term effect will this feedback have on the recipient?

Feelings are important:

What negative effect will this feedback may have on the recipient's feelings?

Can you tune the feedback to match the recipient’s preferences?

The recommendations presented in this section are summarised and presented in a table of considerations (Table 9.1) that can be reviewed by anyone providing a software engineer with feedback. Anyone providing feedback to a software engineer may find it of use to consider the preferences and the *value set* of the feedback recipient, which will help avoid any undesirable outcomes of the feedback.

9.6 LIMITATIONS

Bias may affect the outcome of the results. Sackett (1979) identified six different stages during research where bias could occur, and catalogued 65 different biases. To reduce the potential for bias, at all stages of the research experienced researchers were involved in discussions to help ensure the appropriateness of the design, the reliability of the collected data, and the robustness of any interpretations made from the data. By these checks, the potential for bias is reduced, improving the reliability and validity of the findings.

Caution should be taken when generalising the findings of this research and drawing conclusions applicable to all software engineers. The thoughts, perceptions, reflections and reactions of over 190 software engineers working in different environments, developing different software using a range of tools and methodologies were canvased. Good coverage by a relevant group of software engineers make the findings loosely representative, but as some specialisms may not be covered and targeted sampling was not used, the findings are not strictly representative: the results can be received in some confidence but require caution in use.

The findings of this research are not considered as being representative of all software engineers. It’s evident in the findings that while software engineers typically reported certain perceptions and reactions, in most instances there were still examples of engineers who clearly presented alternative perceptions and reactions. Given this evidence of variation, it is anticipated that variation would exist in a larger sample, further supporting that feedback, and how it is experienced, varies for each individual. Currently, these results should be considered indicative, and provide an insight into the range of thoughts, feelings and reactions of software engineers. Future research could look to repeat and support the findings of this research, including the variation in responses provided by the software engineers, with a larger sample size and using a targeted sampling method to provide wider coverage.

The findings of this research illustrate the importance of feedback characteristics including the impact of the feedback source. However, much of the research investigating the effect of the ‘source’ feedback characteristic used a one-dimensional representation of a source to identify or define the source, such as ‘line manager’ or ‘experienced software engineer’. These single constructs do not fully explore all possible aspects of the source, with the personal connection between the feedback recipient and the feedback source being removed. The state of knowledge prior to each stage of research influenced the research design, causing the personal connection between the source and the participant to be removed during subsequent studies. Further research should use these findings to inform a future study and address this limitation by re-introducing the personal connection between the feedback source and the feedback recipient while investigating the effect of received feedback.

This research identified a range of different aspects reported by software engineers as affected by feedback received in software engineering environments. However, the severity of the impact of received feedback is not addressed or investigated. In instances of feedback where the same aspects were reported as being affected, there is no method able to distinguish between the two instances or identify if one instance was more influential or important than the other. The severity of the impact of received feedback may help to further identify important feedback characteristics. Identifying the severity of the impact of received feedback will provide an additional distinction to evaluate the effect of different forms of feedback where the same aspects were reported as being affected.

The aim of this research was to understand how feedback occurred in software engineering, to clarify how feedback is defined by software engineers, and to investigate the affect feedback can have on the motivation of software engineers. The findings of this research should be viewed as indicative of the thoughts, feelings, perceptions, reflections and reactions of software engineers working in software engineering environments. However, the findings are not precise enough yet to support prediction of the outcome of a specific instance of feedback in the workplace.

CHAPTER 10

CONCLUSION

10.1 FINAL SUMMARY

This research has answered the overall research question: *How does feedback impact the motivation of software engineers?* The findings from this research suggest that feedback is an integral factor in the motivation of software engineers, and based on empirical studies with experienced software engineers, requires adequate consideration in both studies that investigate motivation in software engineering, and in relevant theories of motivation. This research was driven by the lack of previous studies that focused on the impact of feedback on the motivation of software engineers. Investigating feedback and its effect on motivation in software engineering required an overall research design that focused on eliciting the thoughts, reflections, perceptions and reactions of experienced software engineers.

Focused research questions emerged during the literature review and as the research progressed. These emergent questions were answered, and combined to address the overall research question. The following paragraphs address the findings relating to each question, and included in Figure 10.1 is visual representation of where each research question was addressed in this thesis.

Q1 – Do software engineers report feedback as commonly occurring in software engineering environments?

They do (chapter 3). Software engineers reported that they typically knew how their work was progressing through received feedback (chapter 5).

Q2 – How is feedback defined by software engineers?

Software engineers describe feedback in terms of ten characteristics (chapter 3).

Q3 – What are the characteristics of feedback in software engineering environments?

There are 10 characteristics used by software engineers when discussing feedback (chapter 3):

- **Source:** the person or machine sending the feedback, examples: colleague, manager, code compiler.
- **Goal:** the intention of the feedback, examples: reduce stress, remove tension, encourage.
- **Medium:** how the feedback is communicated, examples: verbally, email, phone, body language.
- **Direction:** who the feedback is going to/from, examples: from one person to me, from multiple people to me, from me to one person.
- **Instigation:** the prompt to provide feedback, examples: end of a project, arising issues, annual review.
- **Setting:** the contextual environment when receiving feedback, examples: casual chat, individual meeting, team meeting.
- **Timeliness:** the time difference between the instigation and the sending of feedback, examples: instant, minutes, months.
- **Content:** the polarity and topic of the feedback, examples: positive feedback about task performance, negative feedback about attitude, negative feedback about progress.
- **Preparation:** what needs to be done prior to sending the feedback, examples: producing a document, compiling data.
- **Recipient:** the person/s receiving the feedback, examples: team, individual, division.

Q4 – What forms of feedback do software engineers report receiving?

The analysis of instances of feedback collected during the diary study (chapter 5) found that software engineers receive a range of different forms of feedback. From 76 instances of feedback reported by software engineers in the diary study, 47 distinct combinations of the values of feedback characteristics were identified. However, identifying a sub-set of feedback characteristics found some repeatedly-occurring forms of feedback. Software engineers reported receiving positive feedback more frequently from their peers and their line manager than from a senior manager. During the diary study, software engineers reported experiencing both positive and negative feedback, and feedback was reported as being received during a meeting, during a casual chat, verbally and by email. Software engineers infrequently received feedback from users of software they had produced or from their CEO, but this varied depending on the role of the software engineer.

Q5 – What is the initial effect of received feedback?

Software engineers reported that receiving feedback could affect their feelings, job satisfaction, attitude, behaviour, motivation and performance (chapters 3, 5, 6 and 7).

Q6 – What is the delayed effect of received feedback?

Software engineers reported during the diary study (chapter 5) that feedback they had received during a working day could affect their attitude, behaviour, job satisfaction, motivation and productivity at the end of their working day. The engineers reported that the effect on the listed aspects could be positive or negative.

Q7 – Are there any feedback characteristics that change the impact of received feedback?

The findings of the scenario study (chapter 6) and online survey (chapter 7) indicate that the source, medium and content feedback characteristics can change the impact of received feedback. The findings of a scenario study (chapter 6) and online survey (chapter 7) found that the polarity of the content feedback characteristics indicated a different effect. Positive feedback was reported as affecting job satisfaction, and negative feedback was reported as affecting behaviour and causing the software engineers to want to “fix the problem”. Software engineers reported preferences to the medium used for feedback that they receive (chapters 6 and 7), discussed in response to Q8 below. Software engineers indicated that how valuable they considered the received feedback and the impact of received feedback was affected by the source of the feedback (chapters 6 and 7), discussed in response to Q9 below.

Q8 – What is the effect of a change in feedback medium?

The online survey (chapter 7) found that 50% of the participants reported that receiving feedback through a different medium would change the impact of the feedback. Of the respondents who reported that a change in the medium would change the impact of the feedback, 80% indicated that the feedback impact change would be ‘minor’ or ‘moderate’, and 20% indicated that the feedback impact change would be ‘major’.

Q9 – What is the effect of a change in feedback source?

The online survey (chapter 7) found that feedback received from different sources was reported by software engineers as having a different impact. In the online survey (chapter 7) software engineers reported that feedback from an experienced or senior software engineer was more valuable than feedback from a new software engineer, a line manager, or a senior manager. However, feedback from a line manager or senior manager was reported as affecting more aspects than feedback from an experienced or senior software engineer.

Q10 – Why does the value of a feedback characteristic change the impact of received feedback?

Findings (chapters 3, 5, 6 and 7) suggest that the reason the value of feedback characteristics caused a change in the impact of received feedback is due to the perceptions and individual *value set* of the feedback recipient. A model of feedback in software engineering (chapter 8) identified and discussed this in detail, showing how the perceptions and individual *value set* influence the impact of received feedback. Software engineers assess the value and validity of the source and the content, their preferences for the medium and setting, if the perceived goal of the feedback is acceptable or unacceptable, and if there are any hygiene-factor implications from the received feedback. This assessment results in the impact of the received feedback.

Q11 – How do the findings of this research compare to theories of motivation that identify feedback as a factor?

A review of theories of motivation established that four theories include feedback as a factor (chapter 2). Three of the identified theories of motivation include feedback as a method used to provide the recipient with the knowledge of the results of their actions, and one theory of motivation reports that recognition is typically received verbally through feedback. The findings of this research indicate that theories of motivation do not adequately consider the effect of received feedback, and the effect feedback can have on motivation. Specifically, theories of motivation do not identify the characteristics of feedback, any different forms of feedback, and how different forms of feedback can have a different effect.

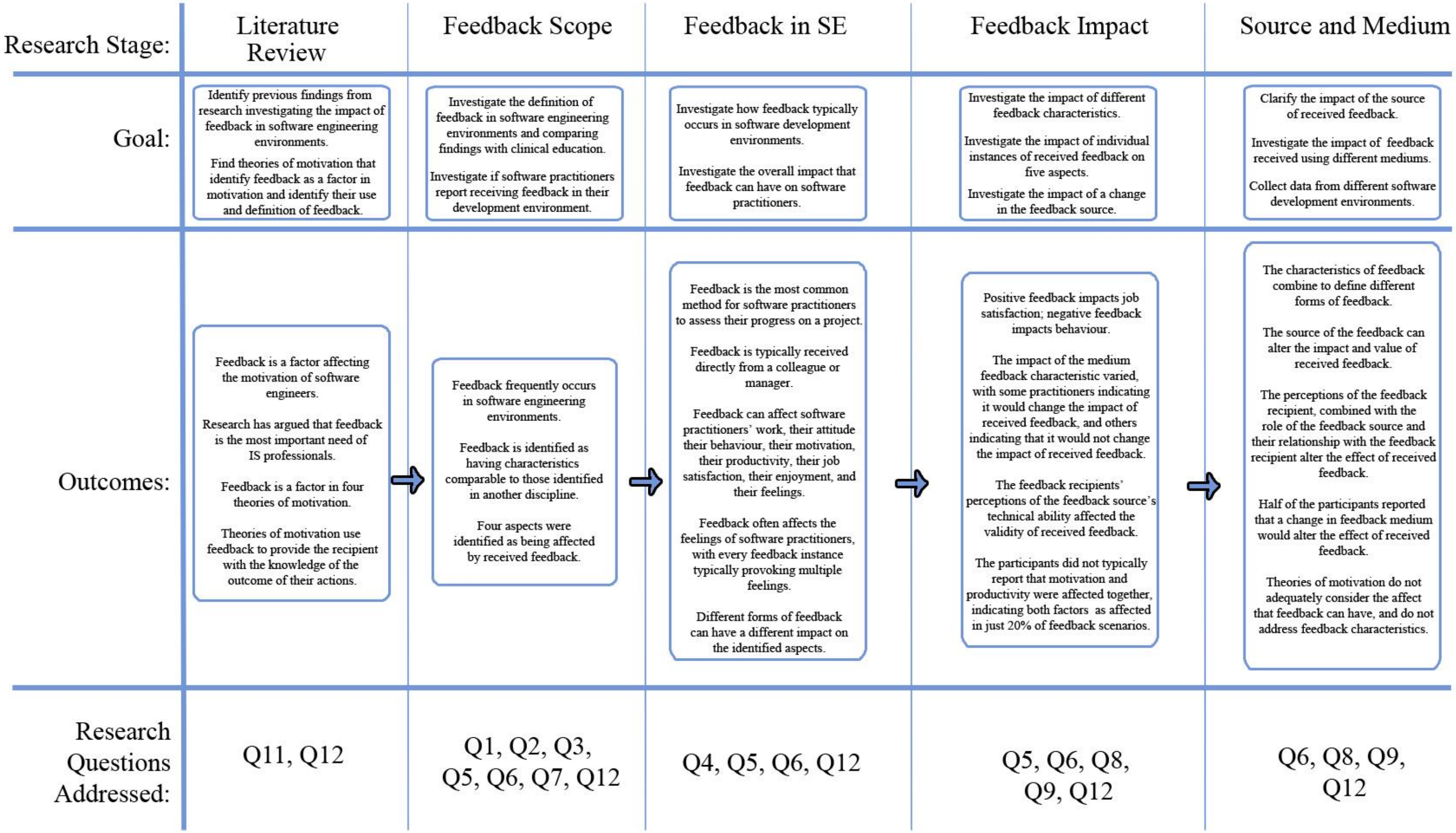
Q12 – What does a model of feedback in software engineering look like?

The findings of the four empirical studies (chapters 3, 5, 6 and 7) and the reviewed literature (chapter 2) combined to develop a model of feedback in software engineering (chapter 8). When feedback is received by a software engineer, they first perceive the characteristics of the feedback, which provides them with knowledge. Their individual *value set* then influences assessment of the validity and value of the feedback source and feedback content, setting and medium preferences, and the feedback goal. Hygiene implications of the received feedback are also considered, before the assessments combine to result in the impact. The impact of the received feedback can be instant or delayed, can be positive or negative, and can affect six different aspects (feelings, motivation, behaviour, job satisfaction, attitude, and performance). Also, received feedback, after it has been perceived and evaluated, may have no impact.

The answers to the twelve focused research questions all contribute to addressing the overall research question of this research:

How does feedback affect the motivation of software engineers?

The findings highlight the important role that feedback has in the motivation of software engineers. The impact feedback can have on software engineers' feelings, attitude, motivation, behaviour, performance and job satisfaction illuminates how influential feedback is in software development environments, and how important it is to provide software engineers with feedback that is both valid and valuable, and is in line with their individual preferences. To ensure that feedback retains its value, the regularity in which it is provided must be monitored, and the *value set* and perceptions of the feedback recipient should be considered if an efficient and healthy feedback provider-recipient relationship is to be maintained.



Research Stage:	Literature Review	Feedback Scope	Feedback in SE	Feedback Impact	Source and Medium
Goal:	Identify previous findings from research investigating the impact of feedback in software engineering environments. Find theories of motivation that identify feedback as a factor in motivation and identify their use and definition of feedback.	Investigate the definition of feedback in software engineering environments and comparing findings with clinical education. Investigate if software practitioners report receiving feedback in their development environment.	Investigate how feedback typically occurs in software development environments. Investigate the overall impact that feedback can have on software practitioners.	Investigate the impact of different feedback characteristics. Investigate the impact of individual instances of received feedback on five aspects. Investigate the impact of a change in the feedback source.	Clarify the impact of the source of received feedback. Investigate the impact of feedback received using different mediums. Collect data from different software development environments.
Outcomes:	Feedback is a factor affecting the motivation of software engineers. Research has argued that feedback is the most important need of IS professionals. Feedback is a factor in four theories of motivation. Theories of motivation use feedback to provide the recipient with the knowledge of the outcome of their actions.	Feedback frequently occurs in software engineering environments. Feedback is identified as having characteristics comparable to those identified in another discipline. Four aspects were identified as being affected by received feedback.	Feedback is the most common method for software practitioners to assess their progress on a project. Feedback is typically received directly from a colleague or manager. Feedback can affect software practitioners' work, their attitude their behaviour, their motivation, their productivity, their job satisfaction, their enjoyment, and their feelings. Feedback often affects the feelings of software practitioners, with every feedback instance typically provoking multiple feelings. Different forms of feedback can have a different impact on the identified aspects.	Positive feedback impacts job satisfaction; negative feedback impacts behaviour. The impact of the medium feedback characteristic varied, with some practitioners indicating it would change the impact of received feedback, and others indicating that it would not change the impact of received feedback. The feedback recipients' perceptions of the feedback source's technical ability affected the validity of received feedback. The participants did not typically report that motivation and productivity were affected together, indicating both factors as affected in just 20% of feedback scenarios.	The characteristics of feedback combine to define different forms of feedback. The source of the feedback can alter the impact and value of received feedback. The perceptions of the feedback recipient, combined with the role of the feedback source and their relationship with the feedback recipient alter the effect of received feedback. Half of the participants reported that a change in feedback medium would alter the effect of received feedback. Theories of motivation do not adequately consider the affect that feedback can have, and do not address feedback characteristics.
Research Questions Addressed:	Q11, Q12	Q1, Q2, Q3, Q5, Q6, Q7, Q12	Q4, Q5, Q6, Q12	Q5, Q6, Q8, Q9, Q12	Q6, Q8, Q9, Q12

Research Stage Key:

Literature - Literature Review, chapter 2
Feedback Scope - Feedback Scoping Study, chapter 3
Feedback in SE - Feedback in One Software Engineering Environment, chapter 5
Feedback Impact - The Short-Term Impact of Received Feedback, chapter 6
Source and Medium - The Effect of 'Source' and 'Medium' Feedback Characteristics, chapter 7

Note: Q10 and Q12 are also addressed during chapter 9, the discussion chapter.

FIGURE 10.1 – IDENTIFICATION OF FINDINGS ADDRESSING ALL RESEARCH QUESTIONS

10.2 CONTRIBUTIONS

This research provides four contributions to knowledge:

1. **Empirical evidence of the effect that feedback has on software engineers.** This research investigated the impact of feedback in software engineering, and identified six aspects that were affected by feedback. This research investigated the effect of different forms of feedback, finding that positive and negative feedback affect different aspects. Software engineers have individual preferences and an individual value set that can alter the impact of received feedback. Overall, this research provides key findings that:
   - Feedback is recognised by software engineers as having characteristics, and that the recognised feedback characteristics are comparable to those identified in clinical education.
   - The values of feedback characteristics, especially of the source, medium and content, can alter the impact of received feedback.
   - The source of feedback reported as the most valuable by software engineers may not be the source of feedback that has the most impact on software engineers.
   - Feedback can affect six aspects of software engineers: attitude, behaviour, motivation, performance, job satisfaction, and feelings.
2. **Implications from this research for theories of motivation that include feedback as a factor relevant to software engineering environments.** The findings of the empirical studies in this research identify that feedback is defined by characteristics, that there are different forms of feedback as identified by the values of their characteristics, and that different forms of feedback have different effects. Theories of motivation do not identify different forms of feedback, or address the affect that different forms of feedback may have. Four theories of motivation could be more suitable for software engineering by considering the effect of different forms of feedback, using the model of feedback in software engineering.
3. **A model of feedback in software engineering.** The findings of this research, combined with relevant literature, underpin the development of a model capturing how feedback is received and reacted to by software engineers.
4. **Recommendations to aid the sender-receiver feedback relationship in software engineering.** From the overall findings of this research several

recommendations can be given to software engineering that will aid the sender-receiver feedback relationship. The recommendations highlight the need for software engineer managers and other sources of feedback to consider the message they wish to communicate in their feedback, and that the individual characteristics and perceptions of the feedback recipient may influence and alter the impact of the received feedback.

10.3 FUTURE WORK

Five potential avenues of future research emerge from the findings of this thesis:

Validation. A model of feedback in software engineering (chapter 8) conceptualises how feedback is received, assessed and reacted to by software engineers. Future research could look to validate the model by implementing it as a computational model that takes inputs of the feedback characteristics, the recipients individual *value set*, and the recipients state of mind / mood / emotions, and then calculates the impact of received feedback. Validating the model may bring to light other areas or aspects of receiving feedback in software engineering that may have not been discovered during this research. Future research would need to establish any other individual *value set* aspects relevant to the model that were not identified in this research, which may not be specific to software engineering and may be identified through a review of relevant literature. This conceptual model would form the basis of an instrument that could be used in the industry to further aid the sender-receiver feedback relationship by establishing the probable impact of received feedback associated with an individual’s characteristics and their state of mind / mood / emotions.

Severity. Future research could investigate the severity of the impact of received feedback. This research focused on identifying aspects impacted by feedback, asking participants to indicate if an aspect was impacted by received feedback. However, this does not consider that while a participant may report the same aspect as impacted for two different instances of feedback, the severity of the impact may be different. Investigating the severity of the impact may provide a more accurate indication of the impact of received feedback, and would provide a further variable that could be used to distinguish the impact of different forms of received feedback.

Source. This research focused on identifying feedback, how it occurred in software engineering environments, and what feedback characteristics were important in affecting the impact of received feedback. The source of the received feedback was one of two main characteristics identified that could change the impact of received feedback. Future research could further investigate the source, and focus on addressing areas of the source not covered in this research. The personal relationships between the feedback recipient and the source, and the feedback recipients’ perceptions of the ability and experience of the source were identified as aspects that could impact the value and validity of received feedback. Future research could investigate this finding, and identify the specific impact that the perceptions of the feedback source by the feedback recipient have on the value of received feedback.

Research Method. This research focused on the thoughts, perceptions, reflections, and reactions of software engineers. The findings present an analysed account of experienced software engineers reporting how they believed feedback affected them in software development environments. A future research avenue could be to attempt to support or dispute these findings through a research design that does not focus specifically on the accounts of software engineers, and instead investigates feedback without relying on the perceptions of software engineers. The findings of this research identified that feedback software engineers believed to be the most valuable may not have the most impact. Investigating any differences between the reports of software engineers as collected during this research, and any other indicators outside of the reports of software engineers as identified by future research, would provide further knowledge about the effect of feedback in software engineering, and may further explain why feedback software engineers believed to be the most valuable may not have the most impact.

Sending Feedback. Future research could take a different approach and look at the opposite side of the sender-receiver feedback relationship, investigating the perceptions of software engineer managers and what they believe the affect is of the feedback they provide to software engineers. This research focused on the perceptions of software engineers, and how received feedback affected them. The findings from this research identified that feedback from line and senior managers provided the widest range of impact on software engineers. Comparing the findings of this study to the impact software engineer managers believe their feedback has may illustrate a gap between what managers believe the take-home message of their feedback is, and what take-home message, if any, is received by software engineers. If completed, the future research could provide valuable knowledge to both software engineers and software engineer managers. The findings of the future research may enable software engineers and software engineer managers to bridge a possible gap between feedback intention and feedback impact, allowing a more efficient feedback exchange that avoids any undesired outcomes including incorrect interpretation and emotional distress.

CONCLUSION

This research revealed the inner workings of feedback in software engineering, and identified how feedback is defined by software engineers. The findings of this research have implications for theories of motivation, and address the suitability in software engineering environments of four motivation theories. This research provides a model of feedback in software engineering that represents how feedback is experienced by software engineers, identifying the effect of each individual's preferences and *value set*. Recommendations are presented that apply the findings of this research to industry environments, which can help to improve the source-recipient feedback relationship that occurs when feedback is given. Overall, from the findings of this research, feedback and the affect that feedback can have on software engineers is better understood; characteristics of feedback that can affect the impact of received feedback have been identified, and the effect of the values of feedback characteristics has been investigated. Five avenues of future research have been identified that can directly build on, or support, the findings of this research.

REFERENCES

Abdel-Hamid, T.K., 1989. A Study of Staff Turnover, Acquisition, and Assimiliation and Their Impact on Software Development Cost and Schedule. Journal of Management Information Systems 6, 21–40.

Agarwal, R., Ferratt, T.W., 2001. Crafting an HR strategy to meet the need for IT workers. Commun. ACM 44, 58–64.

Alaszewski, A., 2006. Using Diaries for Social Research. Thousand Oaks, London; New Delhi: Sage (IQM series).

Altrichter, H., Feldman, A., Posch, P. & Somekh, B., 2008. Teachers investigate their work; An introduction to action research across the professions. Routledge. p. 147. (2nd edition).

Amabile, T.M., Kramer, S.J., 2011. The Progress Principle. Harvard Business Review.

Atkinson, J.W., 1964. An introduction to motivation. Van Nostrand.

Balijepally, V. Mahapatra, R., Nerur, S. 2006. Assessing Personality Profiles of Software Developers in Agile Development Teams. Communications of AIS, Volume 18, Article 4.

Baron, R.A., 1991. Motivation in work settings: Reflections on the core of organizational research. Motivation and Emotion 15, 1–8.

Barrick, M.R., Mount, M.K., 1991. The Big Five Personality Dimensions and Job Performance: A Meta-Analysis. Personnel Psychology, 44

Beecham, S., Baddoo, N., Hall, T., Robinson, H., Sharp, H., 2008. Motivation in Software Engineering: A systematic literature review. Information and Software Technology 50, 860–878.

Berger, M.M., 1994. Feedback in dynamic psychotherapy: Definitions and essential concepts. The American Journal of Psychoanalysis 54, 235–254.

Berridge, K.C., Winkielman, P., 2003. What is an unconscious emotion? (The case for unconscious “liking”). Cognition and Emotion, 17 (2), 181-211.

Black, T.R., 1999. Doing quantitative research in the social sciences: an integrated approach to research design, measurement and statistics. SAGE.

Boehm, B., 1981. Software Engineering Economics. Englewood Cliffs, NJ : Prentice-Hall.

Bouhuys, A. L., Bloem, G. M., & Groothuis, T. G. G. (1995). Induction of depressed and elated mood by music influences the perception of facial emotional expressions in healthy subjects. Journal of Affective Disorders, 33, 215–226.

Broadbent, D.E., 1958. The selective nature of learning, in: Perception and Communication. Pergamon Press, pp. 244–267.

Buchanan, T., Johnson, J.A., Goldberg, L.R., 2005. Implementing a five-factor personality inventory for use on the internet. European Journal of Psychological Assessment 21, 115–127.

Burn, J.M., Daniel Couger, J., Ma, L., 1992. Motivating IT professionals* 1,* 2:: The Hong Kong challenge. Information & Management 22, 269–280.

Capretz, L.F., Ahmed, F., 2010. Why do we need personality diversity in software engineering? SIGSOFT Softw. Eng. Notes 35, 1–11.

Carayon, P., Hoonakker, P., Marchand, S., Schwarz, J., 2003. Job characteristics and quality of working life in the IT workforce: the role of gender, in: Proceedings of the 2003 SIGMIS Conference on Computer Personnel Research: Freedom in Philadelphia--leveraging Differences and Diversity in the IT Workforce. ACM, Philadelphia, Pennsylvania, pp. 58–63.

Cattell, R. B., 1946. The description and measurement of personality. Harcourt, Brace, & World.

Cook, T. D., Campbell, D. T., 1979. Quasi-experimentation: design & analysis issues for field settings. Rand McNally College.

Chen, L.-H., 2008. Job satisfaction among information system (IS) personnel. Computers in Human Behavior 24, 105–118.

Cheney, P.H., 1984. Effects of Individual Characteristics, Organizational Factors and Task Characteristics on Computer Programmer Productivity and Job Satisfaction. Information & Management 7, 209–214.

Claridge, G., 1986. Eysenck’s contribution to the psychology of personality. In: Modgil, S. & Modgil, C. (Eds), Hans Eysenck: Consensus and Controversy (pp. 73-85). London: Falmer Press.

Cohen, L., Manion, L., Morrison, K., 2000. Research Methods in Education, 5th Revised. ed. RoutledgeFalmer.

Couger, J.D., 1989. Comparison of motivating environments for programmer/analysts and programmers in the US, Israel and Singapore. System Sciences, 1989. Vol.IV: Emerging Technologies and Applications Track, Proceedings of the Twenty-Second Annual Hawaii International Conference on 4, 316–323 vol.4.

Couger, J.D., Adelsberger, H., 1988. Environments: Austria compared to the United States. SIGCPR Comput. Pers. 11, 13–17.

Couger, J.D., Ishikawa, A., 1995. Comparing motivation of Japanese computer personnel versus these of the United States. System Sciences, 1995. Vol. IV. Proceedings of the Twenty-Eighth Hawaii International Conference on 4, 1012–1019 vol.4.

Couger, J.D., Ishikawa, A., 1995. Comparing motivation of Japanese computer personnel versus these of the United States. System Sciences, 1995. Vol. IV. Proceedings of the Twenty-Eighth Hawaii International Conference on 4, 1012–1019 vol.4.

Couger, J.D., Zawacki, R.A., 1980. Motivating and Managing Computer Personnel. John Wiley & Sons.

Datamonitor. 2011. Software: Global Industry Guide. Available from: http://www.datamonitor.com/store/Product/software_global_industry_guide?productid=46A43D30-02A3-4425-BFFB-47FCFD5A3AE5, accessed 01/10/2012.

DeMarco, T., and Lister, T., 1999. Peopleware: Productive Projects and Teams. 2nd ed, Dorset House.

Dictionary.com, 2012. Feedback. Retrieved November 12, 2012, from Dictionary.com website: http://dictionary.reference.com/browse/feedback.

Dittrich, J.E., Daniel Couger, J., Zawacki, R.A., 1985. Perceptions of equity, job satisfaction, and intention to quit among data processing personnel. Information & Management 9, 67–75.

Enns, H.G., Ferratt, T.W., Prasad, J., 2006. Beyond stereotypes of IT professionals: implications for IT HR practices. Communications of the ACM 49, 109.

Eysenck, H. J., 1992. Four ways five factors are not basic. Personality and Individual Differences 13 (8): 667–673.

Feder, E.B., Kelly, J., 1986. Education and economic impact: The human processing model. Education 106, 281–290.

Fink, A., 2002. How to Design Survey Studies. Sage Publications.

Fiske, D.W., 1949. Consistency of the factorial structures of personality ratings from different sources. Journal of Abnormal Social Psychology, Vol. 44, pp329-344.

Fowler, J., Floyd, J., 1995. Improving survey questions: Design and evaluation. (Vol. 38). Thousand Oaks, CA: Sage Publications.

França, A. C. C., Gouveia, T.B., Santos, P.C.F., Santana, C.A., Da Silva, F.Q.B. 2011. Motivation in Software Engineering: A Systematic Literature Review Update. Proceedings of the 15th International Conference on Evaluation and Assessment in Software Engineering (EASE ‘2011).

Gambill, S.E., Clark, W.J., Wilkes, R.B., 2000. Toward a holistic model of task design for IS professionals. Information and Management 37, 217–228.

Hackman, J. F., Oldham, G.R., 1975. Development of the Job Diagnostic Survey. Journal of Applied Psychology, Vol. 60, 159-170.

Hackman, J. R., Lawler, E. E. III., 1971. Employee Reactions to Job Characteristics. Journal of Applied Psychology, Vol. 55, 259-286.

Hackman, J.R., Oldham, G.R., 1976. Motivation through the design of work: test of a theory* 1. Organizational behavior and human performance 16, 250–279.

Hall, T., Sharp, H., Beecham, S., Baddoo, N., Robinson, H., 2008. What Do We Know about Developer Motivation? IEEE SOFTWARE 92–94.

Harrison, R., Baddoo, N., Barry, E., Biffl, S., Parra, A., Winter, B., Wuest, J., 1999. Directions and methodologies for empirical software engineering research. Empirical Software Engineering: An International Journal, 4(4):405-410.

Hattie, J., Timperley, H., 2007. The Power of Feedback. Review of Educational Research 77, 81 –112.

Herold, D.M., Greller, M.M., 1977. Research Notes. Feedback: The Definition of a Construct. Academy of Management Journal 20, 142–147.

Hertel, G., Niedner, S., Herrmann, S., 2003. Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel. Research Policy 32, 1159–1177.

Herzberg, F., 1987. One more time: How do you motivate employees?. Harvard Business Review, vol. 65 issue 5, pp109-120.

Herzberg, F., Mausner, B., and Snyderman, B., 1959. The Motivation to Work. 2nd ed, John Wiley & Sons Inc.

Howarth, E., Brown, J. A., 1971. Investigation of personality factors in a Canadian context: I. Marker structure in personality questionnaire items. Canadian Journal of Behavioural Science Vol 3(2), 161-173.

Hufford, M.R., Shields, A.L., 2002. Electronic diaries. Applied Clinical Trials 11, 46–52.

Igbaria, M., Meredith, G., Smith, D.C., 1995. Career orientations of information systems employees in South Africa. Journal of Strategic Information Systems 4, 319–340.

James, W., 1884. What is an emotion? Mind, 9, 118-205.

Jenkins, G. D.,Nadler Jr., D. A. , Lawler E. E., III, Camman, C., 1975. Standardized Observations: An Approach to Measuring the Nature of Jobs. Journal of Applied Psychology, Vol. 60, 171-181.

Kehoe, C.M., Pitkow, J.E. & Morton, K., 1997. Eighth WWW user survey. Available: http://www.gvu.gatech.edu/user_surveys/survey-1997-10.

Khalil, O.E.M., Zawacki, R.A., Zawacki, P.A., Selim, A., 1997. What motivates Egyptian IS managers and personnel: some preliminary results, in: Proceedings of the 1997 ACM SIGCPR Conference on Computer Personnel Research. ACM, San Francisco, California, United States, pp. 187–196.

Kim, S., Wright, B.E., 2007. IT Employee Work Exhaustion. Review of Public Personnel Administration 27, 147 –170.

Kline, P., Barrett, P., 1983. The factors in personality questionnaires among normal subjects. Advances in Behavior Research and Therapy. 5, 141-202.

Kraiger, K., Billings, R.S., Isen, A.M., 1989. The influence of positive affective states on task perceptions and satisfaction. Organizational Behavior and Human Decision Processes 44, 12–25.

Lazar, J., Feng, D.J.H., Hochheiser, D.H., 2010. Research Methods in Human-Computer Interaction. John Wiley and Sons.

LeDuc, A.L.J., 1980. Motivation of programmers. SIGMIS Database 11, 4 – 12.

Lehman, M. M., Belady, L. A., 1976. A model of large system development. IBM Systems Journal, 15(3): 225-252.

Linderbaum, B.A., Levy, P.E., 2010. The Development and Validation of the Feedback Orientation Scale (FOS). Journal of Management 36, 1372 –1405.

Locke, E.A., 1968. Toward a theory of task motivation and incentives* 1. Organizational behavior and human performance 3, 157–189.

Mak, B.L., Sockel, H., 2001. A confirmatory factor analysis of IS employee motivation and retention. Information & Management 38, 265–276.

McClelland, D., 1961. The achieving society. Princeton, N.J, Van Nostrand.

McCrae, R.R., Costa Jr., P.T., 1989. Reinterpreting the Myers-Briggs Type Indicator From the Perspective of the Five-Factor Model of Personality. Journal of Personality 57, 17–40.

McCrae, R.R., John, O.P., 1998. An introduction to the five-factor model and its applications. Personality: critical concepts in psychology 60, 295.

McLean, E.R., Smits, S.J., Tanner, J.R., 1996. The importance of salary on job and career attitudes of information systems professionals. Information & Management 30, 291–299.

Myers, I.B., 1998. Introduction to Type: A Guide to Understanding Your Results on the Myers-Briggs Type Indicator, 6th ed. Center for Applications of myPersonality database. 2011 .Available from: http://www.psychometrics.ppsis.cam.ac.uk/page/255/mypersonality-database.htm, accessed 10/06/2011.

Newell, A., Herbert, A.S., 1972. Human Problem Solving. Prentice-Hall, Englewood Cliffs, NJ.

Norman, W. T., 1963. Toward an adequate taxonomy of personality attributes: Replicated factor structure in peer nomination personality ratings. Journal of Abnormal and Social Psychology, 66, 574-583.

Oppenheim, A. N., 2000. Questionnaire Design, Interviewing and Attitude Measurement. Continuum International Publishing Group.

Paasivaara, M., Lassenius, C., 2003. Collaboration practices in global inter-organizational software development projects. Software Process: Improvement and Practice 8, 183–199.

Packard, M., Cahill, L.,&McGaugh, J. (1994). Amygdala modulation of hippocampal-dependent and caudate nucleus-dependent memory processes. Proceedings of the National Academy of Sciences, USA, 91, 8477–8481., n.d.

Paunonen, S. V., Jackson, D.N., 2000. What Is Beyond the Big Five ? Plenty!. Journal of Personality 68: 821–835.

Phelps, E.A., Ling, S., Carrasco, M., 2006. Emotion facilitates perception and potentiates the perceptual benefits of attention. Psychological Science 17, 292–299.

Procaccino, J.D., Verner, J.M., Shelfer, K.M., Gefen, D., 2005. What do software engineers really think about project success: an exploratory study. Journal of Systems and Software 78, 194–203.

Ramaprasad, A., 1983. On the Definition of Feedback. Behavioral Science, 28:1, 4-13.

Rasch, R.H., Tosi, H.L., 1992. Factors Affecting Software Developers' Performance: An Integrated Approach. MIS Quarterly 16, 395–413.

Riener, C.R., Stefanucci, J.K., Proffitt, D.R., Clore, G., 2011. An effect of mood on the perception of geographical slant. Cognition & Emotion 25, 174–182.

Robertson, I., Smith, M., 1985. Motivation and job design: Theory, research, and practice. St. Paul: West Publishing Co.

Robson, C., 2002. Real world research: A resource for social scientists and practitioner researchers. 2nd ed. Oxford: Blackwell.

Rugg, G., Petre, M., 2006. A Gentle Guide to Research Methods, 1st ed. Open University Press.

Ryan, R.M., Deci, E.L., 2000. Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. Contemporary Educational Psychology 25, 54–67.

Sackett, D.L., 1979. Bias in analytic research. Journal of chronic diseases 32, 51–63.

Salgado, J.F., 1997. The five-factor model of personality and job performance in the European Community. Journal of Applied Psychology, 82, 30-43.

Schwaber, K., and Beedle, M., 2002. Agile Software Development with Scrum. Prentice Hall.

Sells, S. B., Demares, R. G. & Will, D. P. (1970). Dimensions of personality: 1. Congruent factor structure of Guilford and Cattell trait markers. Mulriuariare Behavior Research, 5, 391-422.

Shadish, W. R. Cook, T. D., Campbell, D. T., 2002. Experimental and Quasi-Experimental Designs for Generalized Causal Inference. Houghton Mifflin Company, Boston, New York.

Shanab, M.E., Peterson, D., Dargahi, S., Deroian, P., 1981. The effects of Positive and Negsative Verbal Feedback on the Intrinsic Motivation of Male and Female Subjects. The Journal of Social Psychology, 115, 195-205.

Sharp, H., Baddoo, N., Beecham, S., Hall, T., Robinson, H., 2009. Models of motivation in software engineering. Information and Software Technology 51, 219–233.

Sharp, H., Hall, T., 2009. An initial investigation of software engineers' motivation, in: Cooperative and Human Aspects on Software Engineering, 2009. CHASE '09. ICSE Workshop On. Presented at the Cooperative and Human Aspects on Software Engineering, 2009. CHASE '09. ICSE Workshop on, pp. 84–91.

Smith, C.B., 1997. Casting the net: surveying an Internet population. Journal of Computer Mediated Communication.

Stenmark, D., 2000. The role of intrinsic motivation when managing creative work. Management of Innovation and Technology, 2000. ICMIT 2000. Proceedings of the 2000 IEEE International Conference on 1, 310–315 vol.1.

Stone, A.A., Shiffman, S., Schwartz, J.E., Broderick, J.E., Hufford, M.R., 2002. Patient non-compliance with paper diaries. BMJ 324, 1193 –1194.

Storbeck, J., Clore, G.L., 2005. With Sadness Comes Accuracy; With Happiness, False Memory: Mood and the False Memory Effect. Psychological Science 16, 785–791.

Strelau, J., Eysenck, H.J., 1987. Personality Dimensions and Arousal. Springer.

Tupes, E. C. & Christal, R. E. (1961). Recurrent personality factors based on trait ratings. USAFASD Technical Report, No. 61-97.

Van de Ridder, J.M.M., Stokking, K.M., McGaghie, W.C., Ten Cate, O.T.J., 2008. What is feedback in clinical education? Medical Education 42, 189–197.

de Vaus, D. A., 1991. Surveys in Social Research. 3rd Edition, Sydney, Allen & Unwin Australia and University of London Press.

Voas, J., 2001. The pitfalls of managing a superstar. IT Professional 3, 65–67.

Wegge, J., Haslam, S.A., 2005. Improving work motivation and performance in brainstorming groups: The effects of three group goal-setting strategies. European Journal of Work and Organizational Psychology 14, 400–430.

Wegge, J., Haslam, S.A., 2005. Improving work motivation and performance in brainstorming groups: The effects of three group goal-setting strategies. European Journal of Work and Organizational Psychology 14, 400–430.

Wiener, N., 1950. The Human Use of Human Beings. The Riverside Press (Houghton Mifflin Co.).

Yun, G.W., Trumbo, C.W., 2000. Comparative response to a survey executed by post, e-mail, & web form. Journal of Computer Mediated Communication.

Zawacki, R.A., 1992. Motivating the IS people of the future. Information systems management (Inf. syst. manage.) 9, 73–75.

Zeelenberg, R., Wagenmakers, E.J., Rotteveel, M., 2006. The Impact of Emotion on Perception Bias or Enhanced Processing? Psychological Science 17, 287–291.

APPENDICES

APPENDIX A SECTION 1 – FEEDBACK FOCUS STUDY INTERVIEW

Introduction:

- Who I am
    - Rien Sach – PhD Research Student at the Open University
- What I am researching
    - Feedback and its impact on the motivation of software engineers
- How this will help
    - Provide data to help better understand motivation in our industry, including what's important and what impact it may have
- What they will get
    - An outside perspective on the impact of feedback in your environment
    - Copy of the results of all of the work in this project
    - Pre-publication copies of papers
- Consent form and information form
    - As stated, you have the right to stop and have all the collected data destroyed at any point before
    - Session will be audio recorded if that's OK
    - Please read the information form before we continue
    - Please read and sign the consent form before we continue
- How the rest of the session will go
    - Open discussion on feedback / motivation

Theme 1: Work History

- Are the past 5 projects you listed typical of the types of projects you worked on?
    - If not, how do they differ, or what would be typical?
    - If there is a significant change, why did this change occur?

Theme 2: The different types of feedback + that occur in this person’s workplace

- When I say “feedback”, what does this word mean to you?
- Does feedback occur within your current working environment? (Might be implicit)
- Could you give me some examples of this occurring for you?
- Discussion : Feedback – Aim: Identify types of feedback that occurs
    - Areas to address:
        - Feedback from computers as well as humans
        - Feedback through different mediums
        - Feedback from computers
        - Feedback from peers/supervisors/clients/managers
        - Positive/negative/neutral/informational etc.

Theme 3: The impact of this feedback

- Using the types of feedback already identified, ask how this feedback affects them?
- Discuss the feedbacks identified and their impacts
- After discussing the impacts, discuss these impacts if not already addressed:
        - Continue to be a software engineer
        - Motivation in a role
        - Motivation for a task/project
        - Motivation at a company

Theme 4: Reason for change (relating to feedback)

- Looking back at previous employment changes, can you identify the main causes for these?
- Hypothetically, if the feedback given changed in your current job, do you believe it would have an effect on your willingness to continue working here?
    - Ensure elaboration is given to identify how it would alter willingness to work.
- Discuss this impact through various previous roles/jobs

APPENDIX A SECTION 2 – FEEDBACK SCOPING STUDY PARTICIPANT INTRODUCTION

Research Information

My name is Rien Sach and I am a research student at the Open University. I am researching the impact feedback has on the motivation of software engineers.

In this session we will be discussing your personal experiences of feedback in software engineering environments. During these discussions I am hoping to identify what types of feedback are important to software engineers, and the impact this feedback has on various factors.

While I hope to gather as much data as possible, it's very important that your confidentiality is maintained. With this in mind, all the data collected will only be published and distributed in a way where you will remain anonymous.

While the data will be used as part of my research, the confidentiality of your personal information will be maintained and the data will be securely stored and not distributed without ensuring anonymity.

As my research progresses into further stages, I plan to collect diaries from software engineers writing about their experiences of feedback while at work over a couple of weeks, and also observe feedback occurring in a software engineering environment. All of these different stages will form part of my overall research and provide valuable information on feedback and the impact it has on motivation.

As part of my research at each stage I will be providing the participants with a copy of the results. This data may provide some valuable information to the participants, which will include an outside perspective of the impact of feedback and a thorough analysis of the data.

If at any time before the data has been collated you wish to withdraw from the study and/or have all your data deleted you can do so by requesting this via email.
Thank you for your time
Rien Sach - r.j.sach@open.ac.uk

APPENDIX A SECTION 3 – FEEDBACK SCOPING STUDY DEMOGRAPHIC QUESTIONNAIRE

Hello,

Thank you for agreeing to talk with me about Motivation and Software Engineering. During this short discussion you will be asked questions relating to feedback and motivation in software engineering.

With your permission, I will be audio recording this discussion. It is expected that the discussion will last no longer than 30 minutes. You should have also been given a consent form to read before completing this document.

Before we begin the discussion, please fill in the details below:

Name:

Nationality:

Gender:

Work History (Past 5 projects):

Domain (e.g. Finance, telecoms):	Team Size:	Development Methodology (eg scrum, spiral, tdd):	Role:

Current Company Domain:

Current Company Team Size:

Current Company Methodology:	
Current Company Role:	
Role Details (Duties, typical tasks):	

Age:

< 20 □

21-30 □

31-40 □

41-50 □

51-60 □

61 > □

Experience (Software Engineering):

< 1 years □

1-3 years □

4-8 years □

9-20 years □

20 > years □

Level of Education:

GCSE/O Levels: □

A levels: □

HND: □

Bachelor's Degree: □

Master's Degree: □

Doctorate Degree: □

Other Qualifications (If Applicable):	

Date: ....

Signed: ....

APPENDIX A SECTION 4 – FEEDBACK SCOPING STUDY CONSENT FORM

The Faculty of Maths, Computing and Technology, The Open University.

The role of Feedback in the Motivation of Software Engineers (name of project)

Agreement to Participate

I, (print name)

agree to take part in this research project.

I have had the purposes of the research project explained to me.

I have been informed that I may refuse to participate at any point by simply saying so.

I have been assured that my confidentiality will be protected as specified in the letter/leaflet.

I agree that the information that I provide can be used for educational or research purposes, including publication.

I understand that if I have any concerns or difficulties I can contact:

Rien Sach, Helen Sharp,
Marian Petre

(name of the researcher, and names of supervisors)

at

The Open University
Milton Keynes
MK7 6AA
{r.j.sach; h.c.sharp;
m.petre}@open.ac.uk
+44 (0) 1908 274066

If I want to talk to someone else about this project, I can contact the Associate Dean (Research) at:

Professor Uwe Grimm
The Open University
Milton Keynes
MK7 6AA
MCT-Associate-Dean-Research@open.ac.uk
+44 (0) 1908 274066

Signed:

Date:

APPENDIX B SECTION 1 – FEEDBACK IN ONE SOFTWARE ENGINEERING ENVIRONMENT STUDY INTERVIEW

Introduction:

- Hello I'm Rien Sach – I am a research student at The Open University
- I am researching motivation in software engineering
- This session will help provide data to better understand motivation in our industry, including what is important and what impact it may have
- For your time in this session you will receive a copy of the results from the data collected in this project including pre-publication copies of any papers that contain data from today's session
- As stated in the information and consent form you have the right to stop and have all the collected data destroyed at any point prior to the data being collated.
- Please read the information and consent form and sign it before we continue
- The rest of this session will be an open discussion on motivation in software engineering. I'll be audio recording this session if that's OK?

Theme 1: Demographic info

- So <name>, so you're a Software Engineer at Red Gate
    - What kind of work does being a Software Engineer involve for you?
- Tell me a bit about your background – what's your experience as a software engineer?
    - What kind of projects have you worked on recently?
        - Domain
        - Development Methodology
        - Development team size
        - Typical roles
    - Are these projects typical of your career?
    - How many years have you been a professional software engineer?
    - What did you do before you became a professional software engineer? Qualifications/Education?

Theme 2: Motivation in Software Engineering

- On any of your recent work – tell me what you enjoyed about it?
    - What was good about it?
- What encourages you go that extra mile at work?
    - If they ask "what do you mean extra mile", I mean work harder/longer, be more passionate or interested in your work
- What about any of your recent work that you didn't enjoy as much?
    - What made it less enjoyable for you?
- Is there something that really saps your energy at work?
    - If they ask "what do you mean saps energy", I mean struggle to work harder/longer, be less passionate or interested in your work
- [if relevant] Is there a part of being a software engineer that you prefer doing over other aspects?
    - Why is this?
- [if relevant] And what about the other aspects of your role, what do you like doing the least?
    - Why is this?
- So it's Wednesday morning, middle of the week. You've just woken up, what makes you get up and go to work as a software engineer?
    - When they say money - But that's true for any job, so why as a software engineer?

Theme 3: Feedback and Impact

- What's the communication like with the people you work with?
- How often do you guys talk to each other?

- What kind of stuff do you guys talk about?
    - How do you guys talk about work stuff?
- What happens when you need to ask someone a question while you're at work?
    - Is it always done like this?
    - Are there any other options?
- How do you know when something's going right or wrong?
- How do you know when a colleague thinks you've done a good job?
    - What about if a manager/team leader thinks you've done something good?
- How do you know when a colleague thinks you've think a bad job?
    - What about if a manager/team leader thinks you've done something bad?
- Do you talk to your clients much?
    - What's the communication like?
        - Medium, duration, content etc.
- How does this <feedback> make you feel? (range of examples from current data)
- When you're coding how do you know when something goes right?
    - How does this make you feel?
- When you're coding how do you know when something goes wrong?
    - How does this make you feel?
- What about after working hard all day on some code, and finally at the end of the day you finish the code and get it all working – how does this make you feel?
- What about after working hard all day on some code and at the end of the day you finish and the code is still not working – how does this make you feel?

Check list

- Feedback:
    - Try to identify **informal/formal** feedback if not implicit
    - Feedback from a range of **sources**
    - **How often** do you get this kind of feedback?
    - **Type** of feedback?
    - What kind of **form** does the feedback come in? (**medium**)
    - What kind of **information** is in this feedback? (**subject**)
    - **Where** was this feedback received? (**setting**)

- If not mentioned, ask:
    - **Colleague**
    - **Supervisor**
    - **Computer**
    - **Client**
    - **Other**

APPENDIX B SECTION 2 – FEEDBACK IN ONE SOFTWARE ENGINEERING ENVIRONMENT STUDY INTRODUCTION

Research Information

My name is Rien Sach and I am a research student at the Open University. I am researching the impact feedback has on the motivation of software engineers.

In this session we will be discussing your experiences as a professional software developer. During these discussions I am hoping to explore what motivates software engineers and what impact different factors may have.

While I hope to gather as much data as possible, it is very important that your confidentiality is maintained. With this in mind, all the data collected will be modified to ensure that you will remain anonymous for any publications.

While the data will be used as part of my research, the confidentiality of your personal information will be maintained and the data will be securely stored and not distributed without ensuring anonymity.

As my research progresses I plan to collect diaries from software engineers writing about their experiences while developing software for a couple of weeks, and also observe software engineers in software development environments. All of these different stages will form part of my overall research and provide valuable information on motivation in software engineering and what is important to software engineers.

As part of my research at each stage I will be providing the participants with a copy of the results. This data may provide some valuable information to the participants, which will include an analysis of the data, an outside perspective on motivation, and the important factors that impact the motivation of software engineers.

If at any time before the data has been collated you wish to withdraw from the study and/or have all your data deleted you can do so by requesting this via email.

Thank you for your time
Rien Sach - r.j.sach@open.ac.uk

APPENDIX B SECTION 3 – FEEDBACK IN ONE SOFTWARE ENGINEERING ENVIRONMENT ETHICAL APPROVAL APPLICATION AND APPROVAL NOTIFICATION

Human Participants & Materials Ethics Committee (HPMEC) Project Registration and Risk Checklist

If you are planning a research project that involves human participants (data and/or biological samples), you should complete and submit this checklist so that the HPMEC can decide the level of ethics review that is required. If you have not already done so, refer to the ***OU Ethics Principles for Research involving Human Participants***.

Once you have completed the checklist, save it for your records and email a copy to Research-ethics@open.ac.uk. You will then be contacted by HPMEC regarding the level of ethics review required. No potential participants should be approached to take part in any research until you have submitted your checklist and, where necessary, gained HPMEC approval. Applications for ethics review by HPMEC should be made using the standard proforma.

Section I: Project Details

Project title	The role of Feedback in the Motivation of Software Engineers
Brief description (50 words maximum)	My research aims to catalogue feedback that occurs in software engineering, and then to identify which type of feedback is important and the impact this feedback has. The overall aim will be to identify how important feedback is, why/what makes it important, and how we can use this knowledge.
If your project is externally funded please provide the RED form Reference No.	

Section II: Applicant Details

Name of researcher (applicant)	Rien Sach
Status	Postgraduate Student
Email address	r.j.sach@open.ac.uk
Academic unit	MCT
Ext. no.	**32566**

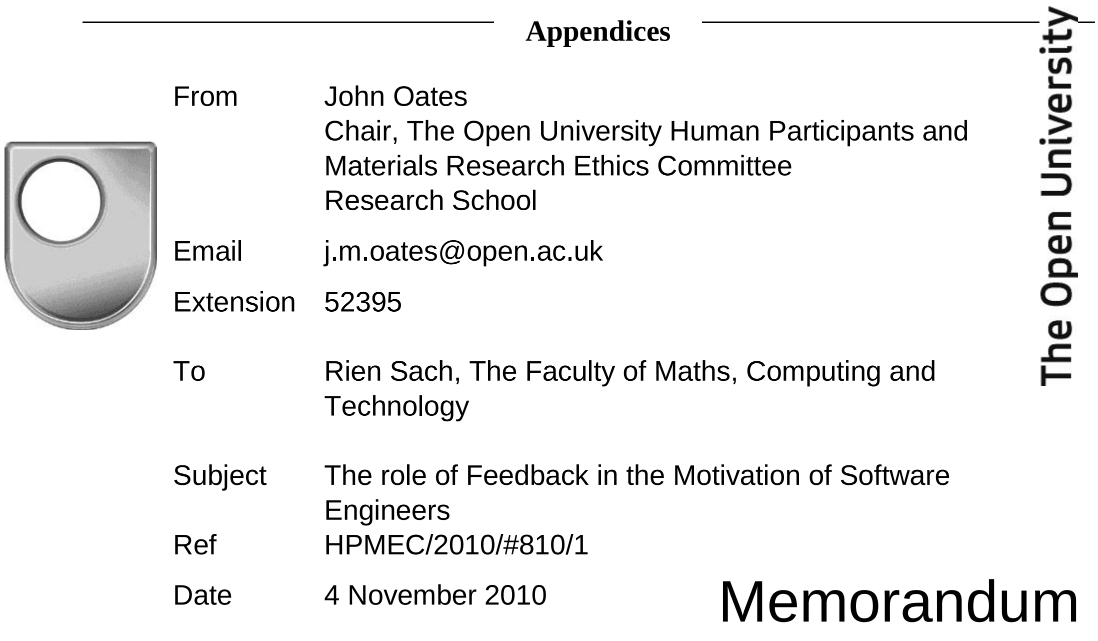
Section III: For students only:

MA/MPhil/PhD/EdD and academic unit:	PhD MCT
Supervisor’s name	Helen Sharp & Marian Petre
Supervisor’s email address	h.c.sharp@open.ac.uk, m.petre@open.ac.uk

Section IV: Risk Checklist

Please answer the following questions:

		Yes	**No**
1	Does the study involve participants who are particularly vulnerable or unable to give informed consent? (e.g. children, people with learning disabilities)	☐	☒
2	Will the study require the co-operation of a gatekeeper for initial access to the groups or individuals to be recruited? (e.g. students at school, members of a self-help group, residents of nursing home)	☐	☒
3	Will it be necessary for participants to take part in the study without their knowledge and consent at the time? (e.g. covert observation of people in non-public places)	☐	☒
4	Will the study involve discussion of sensitive topics (e.g. sexual activity, drug use)?	☐	☒
5	Are drugs, placebos or other substances (e.g. food substances, vitamins) to be administered to the study participants or will the study involve invasive, intrusive or potentially harmful procedure of any kinds?	☐	☒
6	Is pain or more than mild discomfort likely to result from the study?	☐	☒
7	Could the study induce psychological stress or anxiety or cause harm or negative consequences beyond the risks encountered in normal life?	☐	☒
8	Will the study involve prolonged or repetitive testing?	☐	☒
9	Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants?	☐	☒
10	Will the study involve recruitment of patients or staff through the NHS or the use of NHS data?	☐	☒



From John Oates
Chair, The Open University Human Participants and Materials Research Ethics Committee
Research School

Email j.m.oates@open.ac.uk

Extension 52395

To Rien Sach, The Faculty of Maths, Computing and Technology

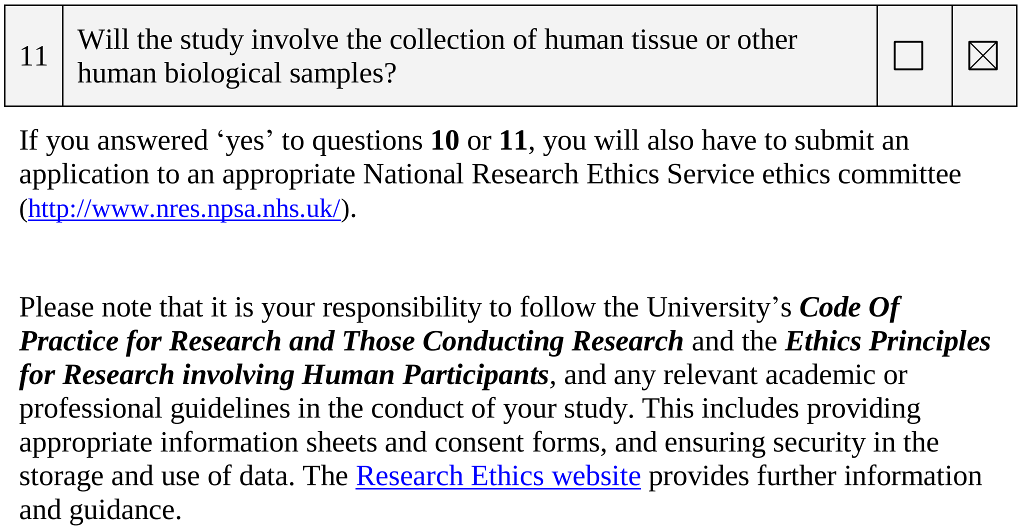
Subject The role of Feedback in the Motivation of Software Engineers

Ref HPMEC/2010/#810/1

Date 4 November 2010

The Open University

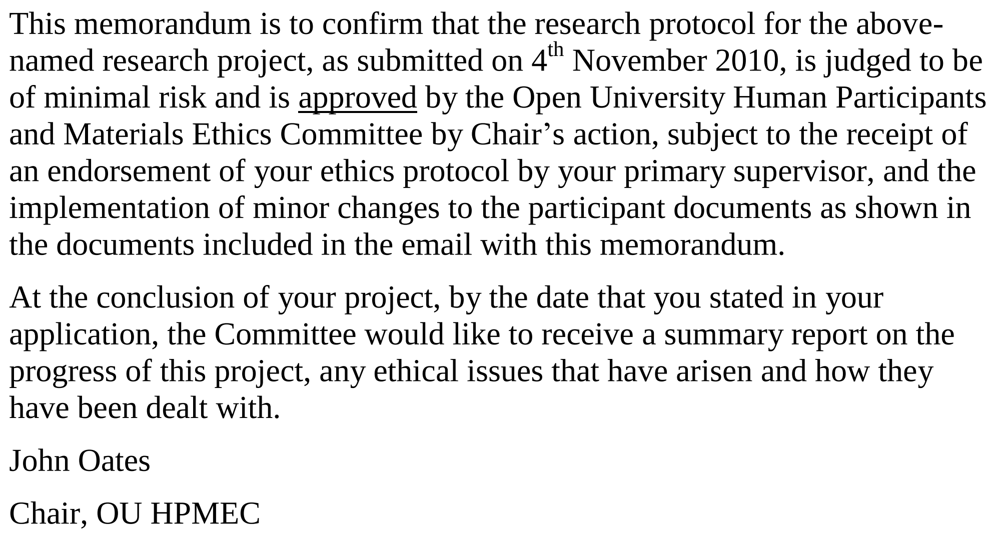
Memorandum



11	Will the study involve the collection of human tissue or other human biological samples?	□	☒

If you answered ‘yes’ to questions **10** or **11**, you will also have to submit an application to an appropriate National Research Ethics Service ethics committee (http://www.nres.npsa.nhs.uk/).

Please note that it is your responsibility to follow the University’s ***Code Of Practice for Research and Those Conducting Research*** and the ***Ethics Principles for Research involving Human Participants***, and any relevant academic or professional guidelines in the conduct of your study. This includes providing appropriate information sheets and consent forms, and ensuring security in the storage and use of data. The Research Ethics website provides further information and guidance.



This memorandum is to confirm that the research protocol for the above-named research project, as submitted on 4th November 2010, is judged to be of minimal risk and is approved by the Open University Human Participants and Materials Ethics Committee by Chair’s action, subject to the receipt of an endorsement of your ethics protocol by your primary supervisor, and the implementation of minor changes to the participant documents as shown in the documents included in the email with this memorandum.

At the conclusion of your project, by the date that you stated in your application, the Committee would like to receive a summary report on the progress of this project, any ethical issues that have arisen and how they have been dealt with.

John Oates

Chair, OU HPMEC

APPENDIX B SECTION 4 – FEEDBACK IN ONE SOFTWARE ENGINEERING ENVIRONMENT STUDY CONSENT FORM

The Faculty of Maths, Computing and Technology, The Open University.

The role of Feedback in the Motivation of Software Engineers (name of project)

Agreement to Participate

I, (print name)

agree to take part in this research project.

I have had the purposes of the research project explained to me.

I have been informed that I may refuse to participate at any point by simply saying so.

I have been assured that my confidentiality will be protected as specified in the letter/leaflet.

I agree that the information that I provide can be used for educational or research purposes, including publication.

I understand that if I have any concerns or difficulties I can contact a member of the project team:

**Rien Sach, Helen Sharp,
Marian Petre**

(name of the researcher, and names of supervisors)

at

**The Open University
Milton Keynes
MK7 6AA
{r.j.sach; h.c.sharp;
m.petre}@open.ac.uk
+44 (0) 1908 274066**

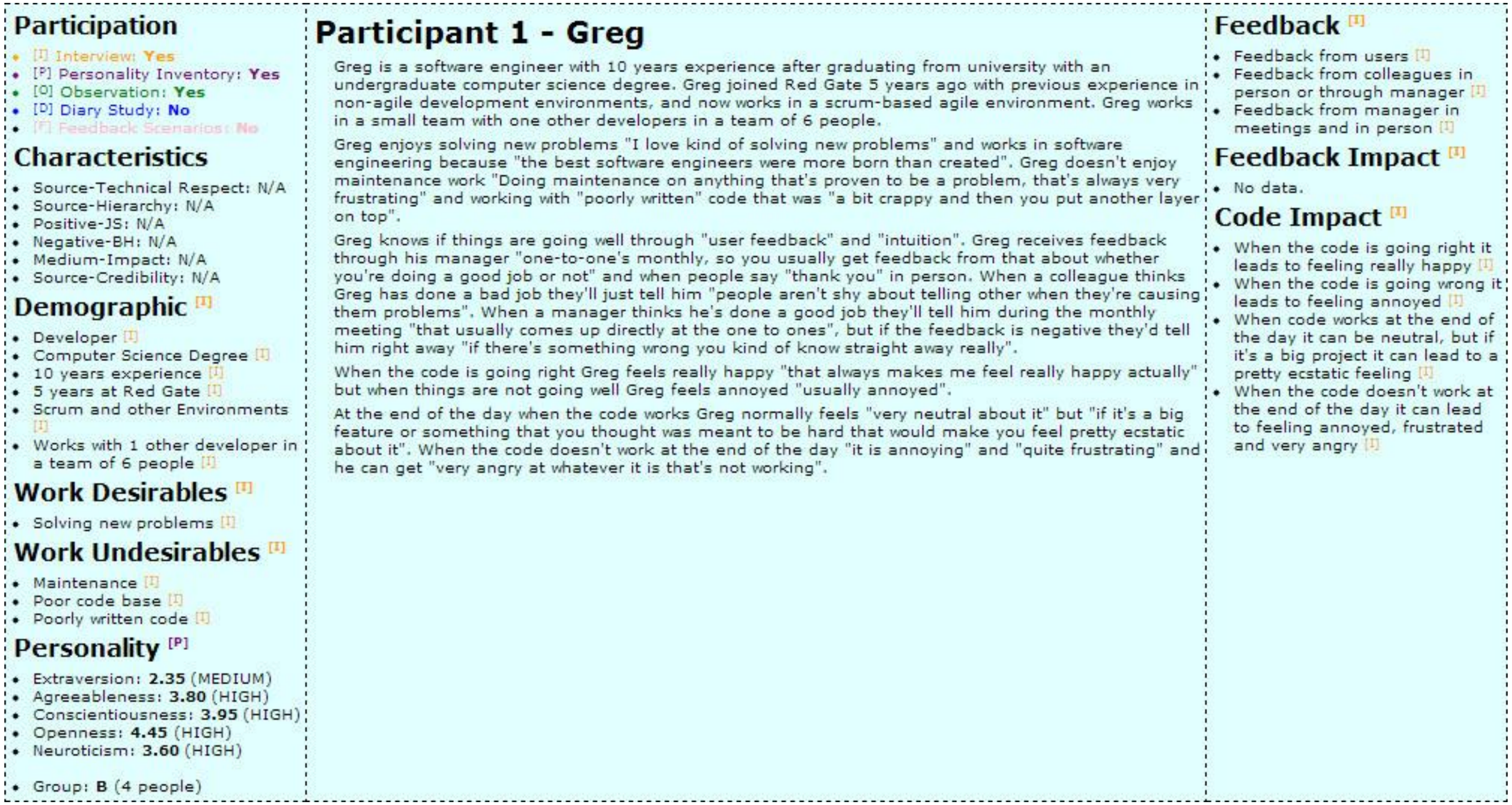
If I want to talk to someone else about this project, I can contact the Associate Dean (Research) at:

**Professor Uwe Grimm
The Open University
Milton Keynes
MK7 6AA
MCT-Associate-Dean-Research@open.ac.uk
+44 (0) 1908 274066**

Signed:

Date:

APPENDIX B SECTION 5 – FEEDBACK IN ONE SOFTWARE ENGINEERING ENVIRONMENT STUDY PERSONAL PROFILES



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **No**
- [F] Feedback Scenarios: **No**

Characteristics

- Source-Technical Respect: N/A
- Source-Hierarchy: N/A
- Positive-JS: N/A
- Negative-BH: N/A
- Medium-Impact: N/A
- Source-Credibility: N/A

Demographic [I]

- Developer [I]
- Computer Science Degree [I]
- 10 years experience [I]
- 5 years at Red Gate [I]
- Scrum and other Environments [I]
- Works with 1 other developer in a team of 6 people [I]

Work Desirables [I]

- Solving new problems [I]

Work Undesirables [I]

- Maintenance [I]
- Poor code base [I]
- Poorly written code [I]

Personality [P]

- Extraversion: **2.35** (MEDIUM)
- Agreeableness: **3.80** (HIGH)
- Conscientiousness: **3.95** (HIGH)
- Openness: **4.45** (HIGH)
- Neuroticism: **3.60** (HIGH)

- Group: **B** (4 people)

Participant 1 - Greg

Greg is a software engineer with 10 years experience after graduating from university with an undergraduate computer science degree. Greg joined Red Gate 5 years ago with previous experience in non-agile development environments, and now works in a scrum-based agile environment. Greg works in a small team with one other developers in a team of 6 people.

Greg enjoys solving new problems "I love kind of solving new problems" and works in software engineering because "the best software engineers were more born than created". Greg doesn't enjoy maintenance work "Doing maintenance on anything that's proven to be a problem, that's always very frustrating" and working with "poorly written" code that was "a bit crappy and then you put another layer on top".

Greg knows if things are going well through "user feedback" and "intuition". Greg receives feedback through his manager "one-to-one's monthly, so you usually get feedback from that about whether you're doing a good job or not" and when people say "thank you" in person. When a colleague thinks Greg has done a bad job they'll just tell him "people aren't shy about telling other when they're causing them problems". When a manager thinks he's done a good job they'll tell him during the monthly meeting "that usually comes up directly at the one to ones", but if the feedback is negative they'd tell him right away "if there's something wrong you kind of know straight away really".

When the code is going right Greg feels really happy "that always makes me feel really happy actually" but when things are not going well Greg feels annoyed "usually annoyed".

At the end of the day when the code works Greg normally feels "very neutral about it" but "if it's a big feature or something that you thought was meant to be hard that would make you feel pretty ecstatic about it". When the code doesn't work at the end of the day "it is annoying" and "quite frustrating" and he can get "very angry at whatever it is that's not working".

Feedback [I]

- Feedback from users [I]
- Feedback from colleagues in person or through manager [I]
- Feedback from manager in meetings and in person [I]

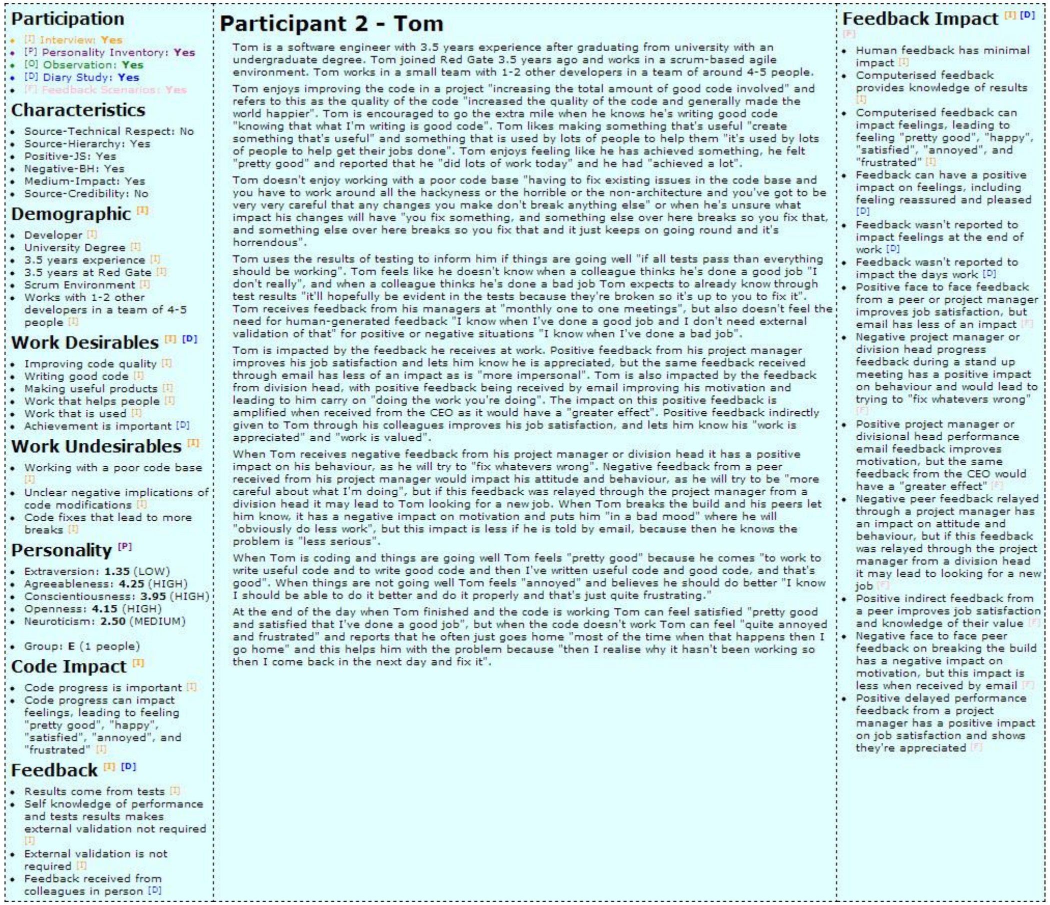
Feedback Impact [I]

- No data.

Code Impact [I]

- When the code is going right it leads to feeling really happy [I]
- When the code is going wrong it leads to feeling annoyed [I]
- When code works at the end of the day it can be neutral, but if it's a big project it can lead to a pretty ecstatic feeling [I]
- When the code doesn't work at the end of the day it can lead to feeling annoyed, frustrated and very angry [I]

PARTICIPANT 1 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **Yes**

Characteristics

- Source-Technical Respect: No
- Source-Hierarchy: Yes
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: Yes
- Source-Credibility: No

Demographic [I]

- Developer [I]
- University Degree [I]
- 3.5 years experience [I]
- 3.5 years at Red Gate [I]
- Scrum Environment [I]
- Works with 1-2 other developers in a team of 4-5 people [I]

Work Desirables [I] [D]

- Improving code quality [I]
- Writing good code [I]
- Making useful products [I]
- Work that helps people [I]
- Work that is used [I]
- Achievement is important [D]

Work Undesirables [I]

- Working with a poor code base [I]
- Unclear negative implications of code modifications [I]
- Code fixes that lead to more breaks [I]

Personality [P]

- Extraversion: **1.35** (LOW)
- Agreeableness: **4.25** (HIGH)
- Conscientiousness: **3.95** (HIGH)
- Openness: **4.15** (HIGH)
- Neuroticism: **2.50** (MEDIUM)

- Group: **E** (1 people)

Code Impact [I]

- Code progress is important [I]
- Code progress can impact feelings, leading to feeling "pretty good", "happy", "satisfied", "annoyed", and "frustrated" [I]

Feedback [I] [D]

- Results come from tests [I]
- Self knowledge of performance and tests results makes external validation not required [I]
- External validation is not required [I]
- Feedback received from colleagues in person [D]

Participant 2 - Tom

Tom is a software engineer with 3.5 years experience after graduating from university with an undergraduate degree. Tom joined Red Gate 3.5 years ago and works in a scrum-based agile environment. Tom works in a small team with 1-2 other developers in a team of around 4-5 people.

Tom enjoys improving the code in a project "increasing the total amount of good code involved" and refers to this as the quality of the code "increased the quality of the code and generally made the world happier". Tom is encouraged to go the extra mile when he knows he's writing good code "knowing that what I'm writing is good code". Tom likes making something that's useful "create something that's useful" and something that is used by lots of people to help them "it's used by lots of people to help get their jobs done". Tom enjoys feeling like he has achieved something, he felt "pretty good" and reported that he "did lots of work today" and he had "achieved a lot".

Tom doesn't enjoy working with a poor code base "having to fix existing issues in the code base and you have to work around all the hackyness or the horrible or the non-architecture and you've got to be very very careful that any changes you make don't break anything else" or when he's unsure what impact his changes will have "you fix something, and something else over here breaks so you fix that, and something else over here breaks so you fix that and it just keeps on going round and it's horrendous".

Tom uses the results of testing to inform him if things are going well "if all tests pass than everything should be working". Tom feels like he doesn't know when a colleague thinks he's done a good job "I don't really", and when a colleague thinks he's done a bad job Tom expects to already know through test results "it'll hopefully be evident in the tests because they're broken so it's up to you to fix it". Tom receives feedback from his managers at "monthly one to one meetings", but also doesn't feel the need for human-generated feedback "I know when I've done a good job and I don't need external validation of that" for positive or negative situations "I know when I've done a bad job".

Tom is impacted by the feedback he receives at work. Positive feedback from his project manager improves his job satisfaction and lets him know he is appreciated, but the same feedback received through email has less of an impact as is "more impersonal". Tom is also impacted by the feedback from division head, with positive feedback being received by email improving his motivation and leading to him carry on "doing the work you're doing". The impact on this positive feedback is amplified when received from the CEO as it would have a "greater effect". Positive feedback indirectly given to Tom through his colleagues improves his job satisfaction, and lets him know his "work is appreciated" and "work is valued".

When Tom receives negative feedback from his project manager or division head it has a positive impact on his behaviour, as he will try to "fix whatevers wrong". Negative feedback from a peer received from his project manager would impact his attitude and behaviour, as he will try to be "more careful about what I'm doing", but if this feedback was relayed through the project manager from a division head it may lead to Tom looking for a new job. When Tom breaks the build and his peers let him know, it has a negative impact on motivation and puts him "in a bad mood" where he will "obviously do less work", but this impact is less if he is told by email, because then he knows the problem is "less serious".

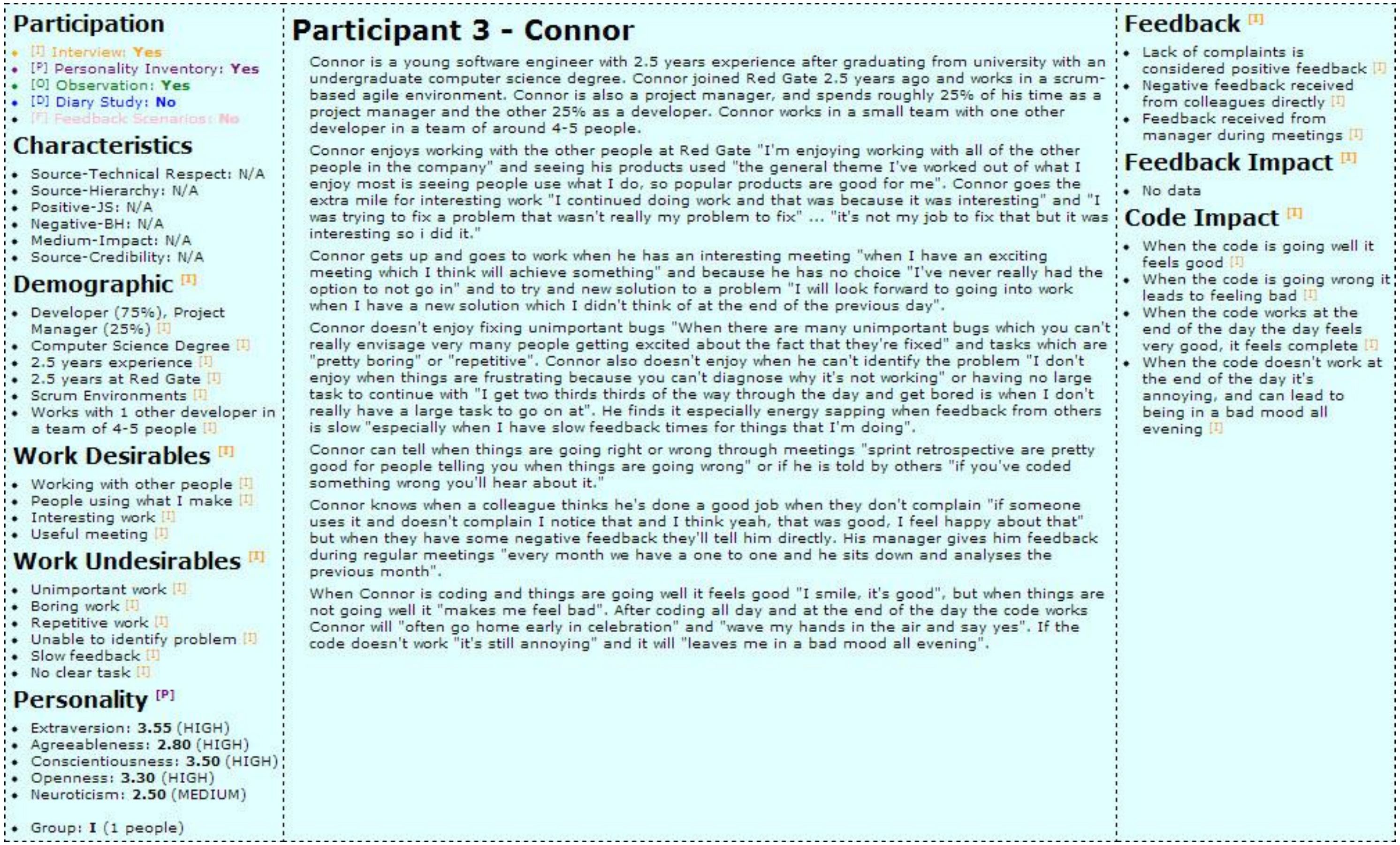
When Tom is coding and things are going well Tom feels "pretty good" because he comes "to work to write useful code and to write good code and then I've written useful code and good code, and that's good". When things are not going well Tom feels "annoyed" and believes he should do better "I know I should be able to do it better and do it properly and that's just quite frustrating."

At the end of the day when Tom finished and the code is working Tom can feel satisfied "pretty good and satisfied that I've done a good job", but when the code doesn't work Tom can feel "quite annoyed and frustrated" and reports that he often just goes home "most of the time when that happens then I go home" and this helps him with the problem because "then I realise why it hasn't been working so then I come back in the next day and fix it".

Feedback Impact [I] [D] [F]

- Human feedback has minimal impact [I]
- Computerised feedback provides knowledge of results [I]
- Computerised feedback can impact feelings, leading to feeling "pretty good", "happy", "satisfied", "annoyed", and "frustrated" [I]
- Feedback can have a positive impact on feelings, including feeling reassured and pleased [D]
- Feedback wasn't reported to impact feelings at the end of work [D]
- Feedback wasn't reported to impact the days work [D]
- Positive face to face feedback from a peer or project manager improves job satisfaction, but email has less of an impact [F]
- Negative project manager or division head progress feedback during a stand up meeting has a positive impact on behaviour and would lead to trying to "fix whatevers wrong" [F]
- Positive project manager or divisional head performance email feedback improves motivation, but the same feedback from the CEO would have a "greater effect" [F]
- Negative peer feedback relayed through a project manager has an impact on attitude and behaviour, but if this feedback was relayed through the project manager from a division head it may lead to looking for a new job [F]
- Positive indirect feedback from a peer improves job satisfaction and knowledge of their value [F]
- Negative face to face peer feedback on breaking the build has a negative impact on motivation, but this impact is less when received by email [F]
- Positive delayed performance feedback from a project manager has a positive impact on job satisfaction and shows they're appreciated [F]

PARTICIPANT 2 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **No**
- [F] Feedback Scenarios: **No**

Characteristics

- Source-Technical Respect: N/A
- Source-Hierarchy: N/A
- Positive-JS: N/A
- Negative-BH: N/A
- Medium-Impact: N/A
- Source-Credibility: N/A

Demographic [I]

- Developer (75%), Project Manager (25%) [I]
- Computer Science Degree [I]
- 2.5 years experience [I]
- 2.5 years at Red Gate [I]
- Scrum Environments [I]
- Works with 1 other developer in a team of 4-5 people [I]

Work Desirables [I]

- Working with other people [I]
- People using what I make [I]
- Interesting work [I]
- Useful meeting [I]

Work Undesirables [I]

- Unimportant work [I]
- Boring work [I]
- Repetitive work [I]
- Unable to identify problem [I]
- Slow feedback [I]
- No clear task [I]

Personality [P]

- Extraversion: **3.55** (HIGH)
- Agreeableness: **2.80** (HIGH)
- Conscientiousness: **3.50** (HIGH)
- Openness: **3.30** (HIGH)
- Neuroticism: **2.50** (MEDIUM)

- Group: **I** (1 people)

Participant 3 - Connor

Connor is a young software engineer with 2.5 years experience after graduating from university with an undergraduate computer science degree. Connor joined Red Gate 2.5 years ago and works in a scrum-based agile environment. Connor is also a project manager, and spends roughly 25% of his time as a project manager and the other 25% as a developer. Connor works in a small team with one other developer in a team of around 4-5 people.

Connor enjoys working with the other people at Red Gate "I'm enjoying working with all of the other people in the company" and seeing his products used "the general theme I've worked out of what I enjoy most is seeing people use what I do, so popular products are good for me". Connor goes the extra mile for interesting work "I continued doing work and that was because it was interesting" and "I was trying to fix a problem that wasn't really my problem to fix" ... "it's not my job to fix that but it was interesting so i did it."

Connor gets up and goes to work when he has an interesting meeting "when I have an exciting meeting which I think will achieve something" and because he has no choice "I've never really had the option to not go in" and to try and new solution to a problem "I will look forward to going into work when I have a new solution which I didn't think of at the end of the previous day".

Connor doesn't enjoy fixing unimportant bugs "When there are many unimportant bugs which you can't really envisage very many people getting excited about the fact that they're fixed" and tasks which are "pretty boring" or "repetitive". Connor also doesn't enjoy when he can't identify the problem "I don't enjoy when things are frustrating because you can't diagnose why it's not working" or having no large task to continue with "I get two thirds thirds of the way through the day and get bored is when I don't really have a large task to go on at". He finds it especially energy sapping when feedback from others is slow "especially when I have slow feedback times for things that I'm doing".

Connor can tell when things are going right or wrong through meetings "sprint retrospective are pretty good for people telling you when things are going wrong" or if he is told by others "if you've coded something wrong you'll hear about it."

Connor knows when a colleague thinks he's done a good job when they don't complain "if someone uses it and doesn't complain I notice that and I think yeah, that was good, I feel happy about that" but when they have some negative feedback they'll tell him directly. His manager gives him feedback during regular meetings "every month we have a one to one and he sits down and analyses the previous month".

When Connor is coding and things are going well it feels good "I smile, it's good", but when things are not going well it "makes me feel bad". After coding all day and at the end of the day the code works Connor will "often go home early in celebration" and "wave my hands in the air and say yes". If the code doesn't work "it's still annoying" and it will "leaves me in a bad mood all evening".

Feedback [I]

- Lack of complaints is considered positive feedback [I]
- Negative feedback received from colleagues directly [I]
- Feedback received from manager during meetings [I]

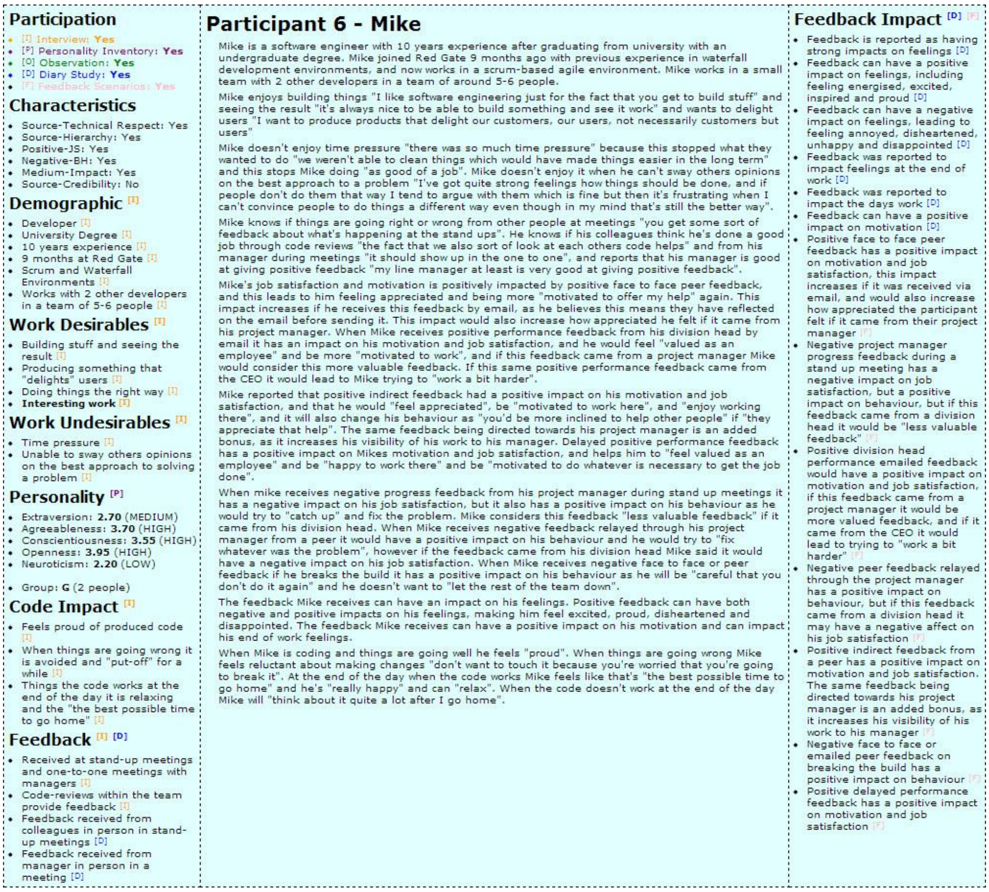
Feedback Impact [I]

- No data

Code Impact [I]

- When the code is going well it feels good [I]
- When the code is going wrong it leads to feeling bad [I]
- When the code works at the end of the day the day feels very good, it feels complete [I]
- When the code doesn't work at the end of the day it's annoying, and can lead to being in a bad mood all evening [I]

PARTICIPANT 3 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **Yes**

Characteristics

- Source-Technical Respect: Yes
- Source-Hierarchy: Yes
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: Yes
- Source-Credibility: No

Demographic [I]

- Developer [I]
- University Degree [I]
- 10 years experience [I]
- 9 months at Red Gate [I]
- Scrum and Waterfall Environments [I]
- Works with 2 other developers in a team of 5-6 people [I]

Work Desirables [I]

- Building stuff and seeing the result [I]
- Producing something that "delights" users [I]
- Doing things the right way [I]
- **Interesting work** [I]

Work Undesirables [I]

- Time pressure [I]
- Unable to sway others opinions on the best approach to solving a problem [I]

Personality [P]

- Extraversion: **2.70** (MEDIUM)
- Agreeableness: **3.70** (HIGH)
- Conscientiousness: **3.55** (HIGH)
- Openness: **3.95** (HIGH)
- Neuroticism: **2.20** (LOW)

- Group: **G** (2 people)

Code Impact [I]

- Feels proud of produced code [I]
- When things are going wrong it is avoided and "put-off" for a while [I]
- Things the code works at the end of the day it is relaxing and the "the best possible time to go home" [I]

Feedback [I] [D]

- Received at stand-up meetings and one-to-one meetings with managers [I]
- Code-reviews within the team provide feedback [I]
- Feedback received from colleagues in person in stand-up meetings [D]
- Feedback received from manager in person in a meeting [D]

Participant 6 - Mike

Mike is a software engineer with 10 years experience after graduating from university with an undergraduate degree. Mike joined Red Gate 9 months ago with previous experience in waterfall development environments, and now works in a scrum-based agile environment. Mike works in a small team with 2 other developers in a team of around 5-6 people.

Mike enjoys building things "I like software engineering just for the fact that you get to build stuff" and seeing the result "it's always nice to be able to build something and see it work" and wants to delight users "I want to produce products that delight our customers, our users, not necessarily customers but users"

Mike doesn't enjoy time pressure "there was so much time pressure" because this stopped what they wanted to do "we weren't able to clean things which would have made things easier in the long term" and this stops Mike doing "as good of a job". Mike doesn't enjoy it when he can't sway others opinions on the best approach to a problem "I've got quite strong feelings how things should be done, and if people don't do them that way I tend to argue with them which is fine but then it's frustrating when I can't convince people to do things a different way even though in my mind that's still the better way".

Mike knows if things are going right or wrong from other people at meetings "you get some sort of feedback about what's happening at the stand ups". He knows if his colleagues think he's done a good job through code reviews "the fact that we also sort of look at each others code helps" and from his manager during meetings "it should show up in the one to one", and reports that his manager is good at giving positive feedback "my line manager at least is very good at giving positive feedback".

Mike's job satisfaction and motivation is positively impacted by positive face to face peer feedback, and this leads to him feeling appreciated and being more "motivated to offer my help" again. This impact increases if he receives this feedback by email, as he believes this means they have reflected on the email before sending it. This impact would also increase how appreciated he felt if it came from his project manager. When Mike receives positive performance feedback from his division head by email it has an impact on his motivation and job satisfaction, and he would feel "valued as an employee" and be more "motivated to work", and if this feedback came from a project manager Mike would consider this more valuable feedback. If this same positive performance feedback came from the CEO it would lead to Mike trying to "work a bit harder".

Mike reported that positive indirect feedback had a positive impact on his motivation and job satisfaction, and that he would "feel appreciated", be "motivated to work here", and "enjoy working there", and it will also change his behaviour as "you'd be more inclined to help other people" if "they appreciate that help". The same feedback being directed towards his project manager is an added bonus, as it increases his visibility of his work to his manager. Delayed positive performance feedback has a positive impact on Mikes motivation and job satisfaction, and helps him to "feel valued as an employee" and be "happy to work there" and be "motivated to do whatever is necessary to get the job done".

When mike receives negative progress feedback from his project manager during stand up meetings it has a negative impact on his job satisfaction, but it also has a positive impact on his behaviour as he would try to "catch up" and fix the problem. Mike considers this feedback "less valuable feedback" if it came from his division head. When Mike receives negative feedback relayed through his project manager from a peer it would have a positive impact on his behaviour and he would try to "fix whatever was the problem", however if the feedback came from his division head Mike said it would have a negative impact on his job satisfaction. When Mike receives negative face to face or peer feedback if he breaks the build it has a positive impact on his behaviour as he will be "careful that you don't do it again" and he doesn't want to "let the rest of the team down".

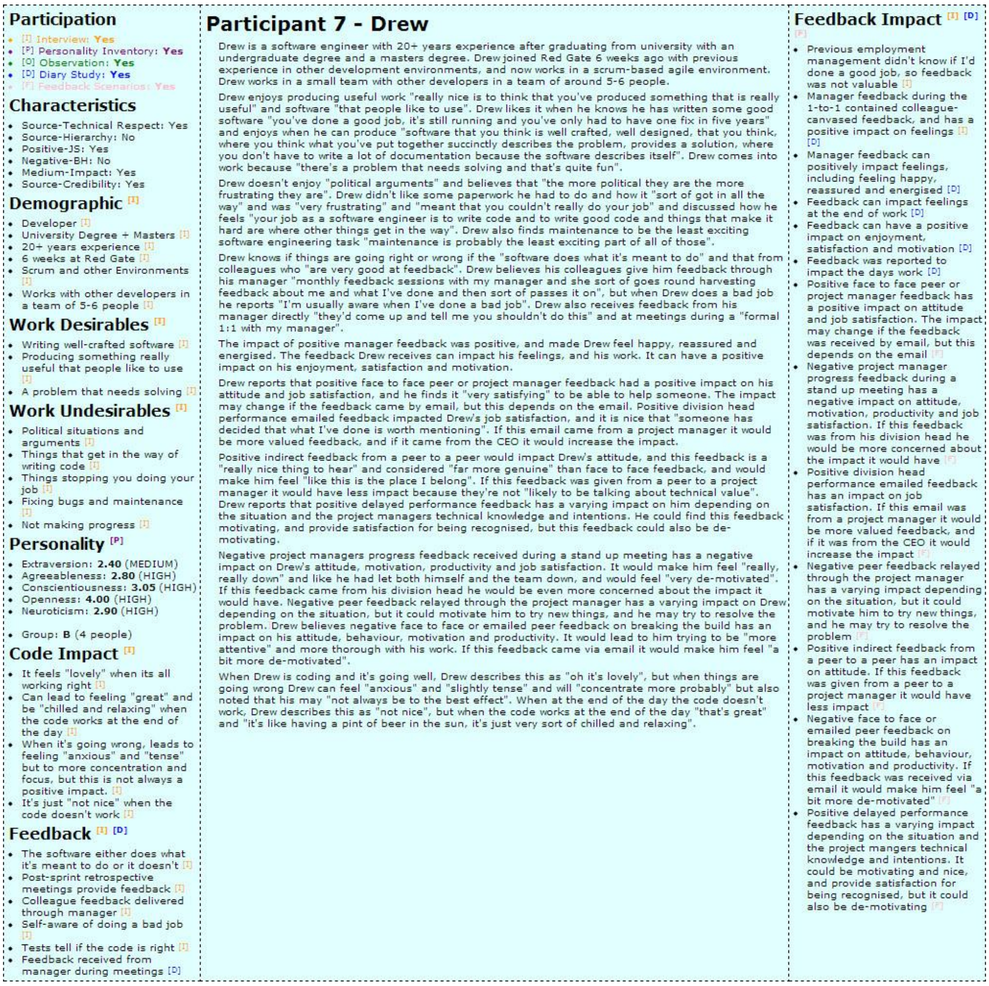
The feedback Mike receives can have an impact on his feelings. Positive feedback can have both negative and positive impacts on his feelings, making him feel excited, proud, disheartened and disappointed. The feedback Mike receives can have a positive impact on his motivation and can impact his end of work feelings.

When Mike is coding and things are going well he feels "proud". When things are going wrong Mike feels reluctant about making changes "don't want to touch it because you're worried that you're going to break it". At the end of the day when the code works Mike feels like that's "the best possible time to go home" and he's "really happy" and can "relax". When the code doesn't work at the end of the day Mike will "think about it quite a lot after I go home".

Feedback Impact [D] [F]

- Feedback is reported as having strong impacts on feelings [D]
- Feedback can have a positive impact on feelings, including feeling energised, excited, inspired and proud [D]
- Feedback can have a negative impact on feelings, leading to feeling annoyed, disheartened, unhappy and disappointed [D]
- Feedback was reported to impact feelings at the end of work [D]
- Feedback was reported to impact the days work [D]
- Feedback can have a positive impact on motivation [D]
- Positive face to face peer feedback has a positive impact on motivation and job satisfaction, this impact increases if it was received via email, and would also increase how appreciated the participant felt if it came from their project manager [F]
- Negative project manager progress feedback during a stand up meeting has a negative impact on job satisfaction, but a positive impact on behaviour, but if this feedback came from a division head it would be "less valuable feedback" [F]
- Positive division head performance emailed feedback would have a positive impact on motivation and job satisfaction, if this feedback came from a project manager it would be more valued feedback, and if it came from the CEO it would lead to trying to "work a bit harder" [F]
- Negative peer feedback relayed through the project manager has a positive impact on behaviour, but if this feedback came from a division head it may have a negative affect on his job satisfaction [F]
- Positive indirect feedback from a peer has a positive impact on motivation and job satisfaction. The same feedback being directed towards his project manager is an added bonus, as it increases his visibility of his work to his manager [F]
- Negative face to face or emailed peer feedback on breaking the build has a positive impact on behaviour [F]
- Positive delayed performance feedback has a positive impact on motivation and job satisfaction [F]

PARTICIPANT 6 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **Yes**

Characteristics

- Source-Technical Respect: Yes
- Source-Hierarchy: No
- Positive-JS: Yes
- Negative-BH: No
- Medium-Impact: Yes
- Source-Credibility: Yes

Demographic [I]

- Developer [I]
- University Degree + Masters [I]
- 20+ years experience [I]
- 6 weeks at Red Gate [I]
- Scrum and other Environments [I]
- Works with other developers in a team of 5-6 people [I]

Work Desirables [I]

- Writing well-crafted software [I]
- Producing something really useful that people like to use [I]
- A problem that needs solving [I]

Work Undesirables [I]

- Political situations and arguments [I]
- Things that get in the way of writing code [I]
- Things stopping you doing your job [I]
- Fixing bugs and maintenance [I]
- Not making progress [I]

Personality [P]

- Extraversion: **2.40** (MEDIUM)
- Agreeableness: **2.80** (HIGH)
- Conscientiousness: **3.05** (HIGH)
- Openness: **4.00** (HIGH)
- Neuroticism: **2.90** (HIGH)

- Group: **B** (4 people)

Code Impact [I]

- It feels "lovely" when its all working right [I]
- Can lead to feeling "great" and be "chilled and relaxing" when the code works at the end of the day [I]
- When it's going wrong, leads to feeling "anxious" and "tense" but to more concentration and focus, but this is not always a positive impact. [I]
- It's just "not nice" when the code doesn't work [I]

Feedback [I] [D]

- The software either does what it's meant to do or it doesn't [I]
- Post-sprint retrospective meetings provide feedback [I]
- Colleague feedback delivered through manager [I]
- Self-aware of doing a bad job [I]
- Tests tell if the code is right [I]
- Feedback received from manager during meetings [D]

Participant 7 - Drew

Drew is a software engineer with 20+ years experience after graduating from university with an undergraduate degree and a masters degree. Drew joined Red Gate 6 weeks ago with previous experience in other development environments, and now works in a scrum-based agile environment. Drew works in a small team with other developers in a team of around 5-6 people.

Drew enjoys producing useful work "really nice is to think that you've produced something that is really useful" and software "that people like to use". Drew likes it when he knows he has written some good software "you've done a good job, it's still running and you've only had to have one fix in five years" and enjoys when he can produce "software that you think is well crafted, well designed, that you think, where you think what you've put together succinctly describes the problem, provides a solution, where you don't have to write a lot of documentation because the software describes itself". Drew comes into work because "there's a problem that needs solving and that's quite fun".

Drew doesn't enjoy "political arguments" and believes that "the more political they are the more frustrating they are". Drew didn't like some paperwork he had to do and how it "sort of got in all the way" and was "very frustrating" and "meant that you couldn't really do your job" and discussed how he feels "your job as a software engineer is to write code and to write good code and things that make it hard are where other things get in the way". Drew also finds maintenance to be the least exciting software engineering task "maintenance is probably the least exciting part of all of those".

Drew knows if things are going right or wrong if the "software does what it's meant to do" and that from colleagues who "are very good at feedback". Drew believes his colleagues give him feedback through his manager "monthly feedback sessions with my manager and she sort of goes round harvesting feedback about me and what I've done and then sort of passes it on", but when Drew does a bad job he reports "I'm usually aware when I've done a bad job". Drew also receives feedback from his manager directly "they'd come up and tell me you shouldn't do this" and at meetings during a "formal 1:1 with my manager".

The impact of positive manager feedback was positive, and made Drew feel happy, reassured and energised. The feedback Drew receives can impact his feelings, and his work. It can have a positive impact on his enjoyment, satisfaction and motivation.

Drew reports that positive face to face peer or project manager feedback had a positive impact on his attitude and job satisfaction, and he finds it "very satisfying" to be able to help someone. The impact may change if the feedback came by email, but this depends on the email. Positive division head performance emailed feedback impacted Drew's job satisfaction, and it is nice that "someone has decided that what I've done is worth mentioning". If this email came from a project manager it would be more valued feedback, and if it came from the CEO it would increase the impact.

Positive indirect feedback from a peer to a peer would impact Drew's attitude, and this feedback is a "really nice thing to hear" and considered "far more genuine" than face to face feedback, and would make him feel "like this is the place I belong". If this feedback was given from a peer to a project manager it would have less impact because they're not "likely to be talking about technical value". Drew reports that positive delayed performance feedback has a varying impact on him depending on the situation and the project managers technical knowledge and intentions. He could find this feedback motivating, and provide satisfaction for being recognised, but this feedback could also be de-motivating.

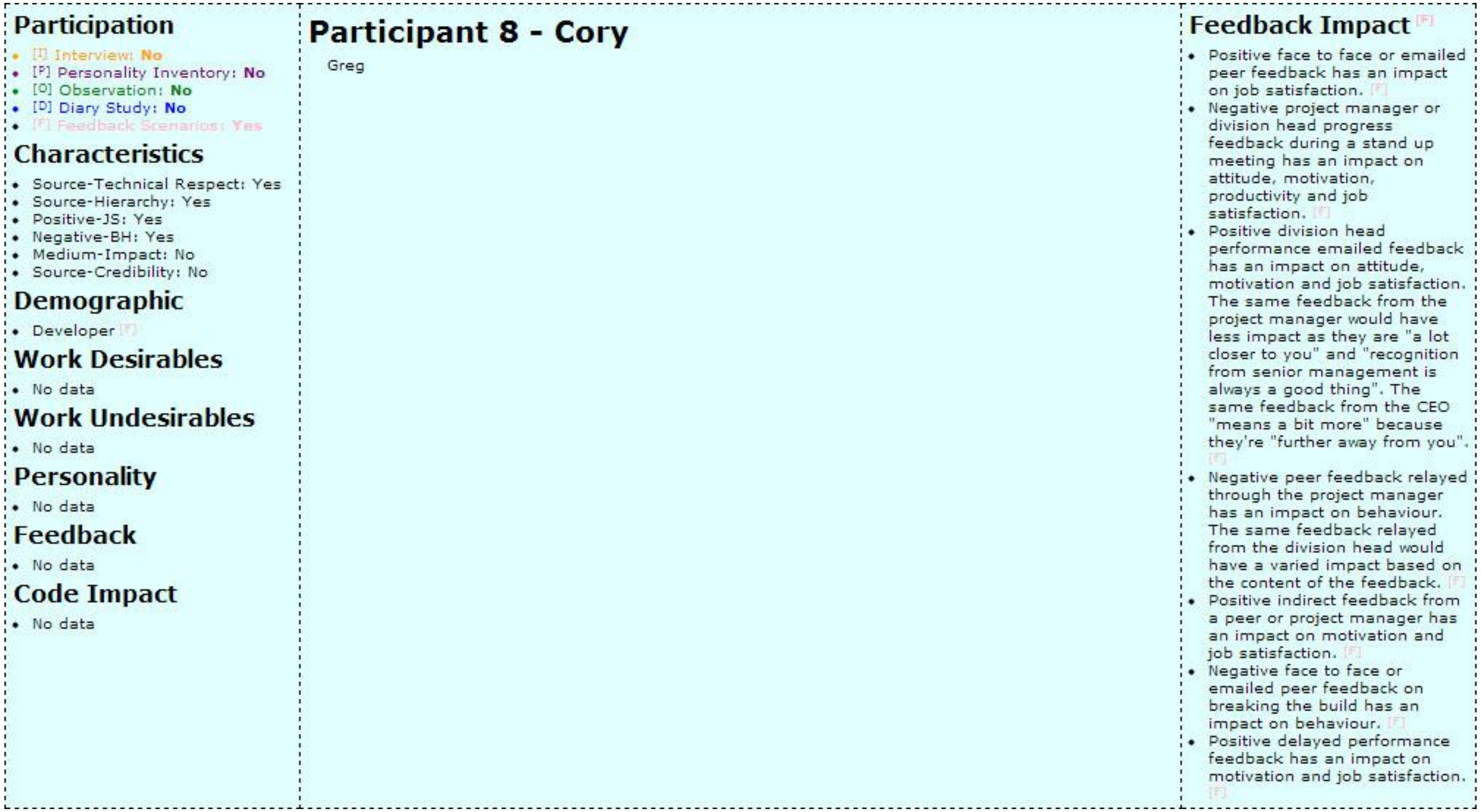
Negative project managers progress feedback received during a stand up meeting has a negative impact on Drew's attitude, motivation, productivity and job satisfaction. It would make him feel "really, really down" and like he had let both himself and the team down, and would feel "very de-motivated". If this feedback came from his division head he would be even more concerned about the impact it would have. Negative peer feedback relayed through the project manager has a varying impact on Drew depending on the situation, but it could motivate him to try new things, and he may try to resolve the problem. Drew believes negative face to face or emailed peer feedback on breaking the build has an impact on his attitude, behaviour, motivation and productivity. It would lead to him trying to be "more attentive" and more thorough with his work. If this feedback came via email it would make him feel "a bit more de-motivated".

When Drew is coding and it's going well, Drew describes this as "oh it's lovely", but when things are going wrong Drew can feel "anxious" and "slightly tense" and will "concentrate more probably" but also noted that his may "not always be to the best effect". When at the end of the day the code doesn't work, Drew describes this as "not nice", but when the code works at the end of the day "that's great" and "it's like having a pint of beer in the sun, it's just very sort of chilled and relaxing".

Feedback Impact [I] [D] [F]

- Previous employment management didn't know if I'd done a good job, so feedback was not valuable [I]
- Manager feedback during the 1-to-1 contained colleague-canvased feedback, and has a positive impact on feelings [I] [D]
- Manager feedback can positively impact feelings, including feeling happy, reassured and energised [D]
- Feedback can impact feelings at the end of work [D]
- Feedback can have a positive impact on enjoyment, satisfaction and motivation [D]
- Feedback was reported to impact the days work [D]
- Positive face to face peer or project manager feedback has a positive impact on attitude and job satisfaction. The impact may change if the feedback was received by email, but this depends on the email [F]
- Negative project manager progress feedback during a stand up meeting has a negative impact on attitude, motivation, productivity and job satisfaction. If this feedback was from his division head he would be more concerned about the impact it would have [F]
- Positive division head performance emailed feedback has an impact on job satisfaction. If this email was from a project manager it would be more valued feedback, and if it was from the CEO it would increase the impact [F]
- Negative peer feedback relayed through the project manager has a varying impact depending on the situation, but it could motivate him to try new things, and he may try to resolve the problem [F]
- Positive indirect feedback from a peer to a peer has an impact on attitude. If this feedback was given from a peer to a project manager it would have less impact [F]
- Negative face to face or emailed peer feedback on breaking the build has an impact on attitude, behaviour, motivation and productivity. If this feedback was received via email it would make him feel "a bit more de-motivated" [F]
- Positive delayed performance feedback has a varying impact depending on the situation and the project mangers technical knowledge and intentions. It could be motivating and nice, and provide satisfaction for being recognised, but it could also be de-motivating [F]

PARTICIPANT 7 PERSONAL PROFILE



Participation

- [I] Interview: **No**
- [P] Personality Inventory: **No**
- [O] Observation: **No**
- [D] Diary Study: **No**
- [F] Feedback Scenarios: **Yes**

Characteristics

- Source-Technical Respect: Yes
- Source-Hierarchy: Yes
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: No
- Source-Credibility: No

Demographic

- Developer [F]

Work Desirables

- No data

Work Undesirables

- No data

Personality

- No data

Feedback

- No data

Code Impact

- No data

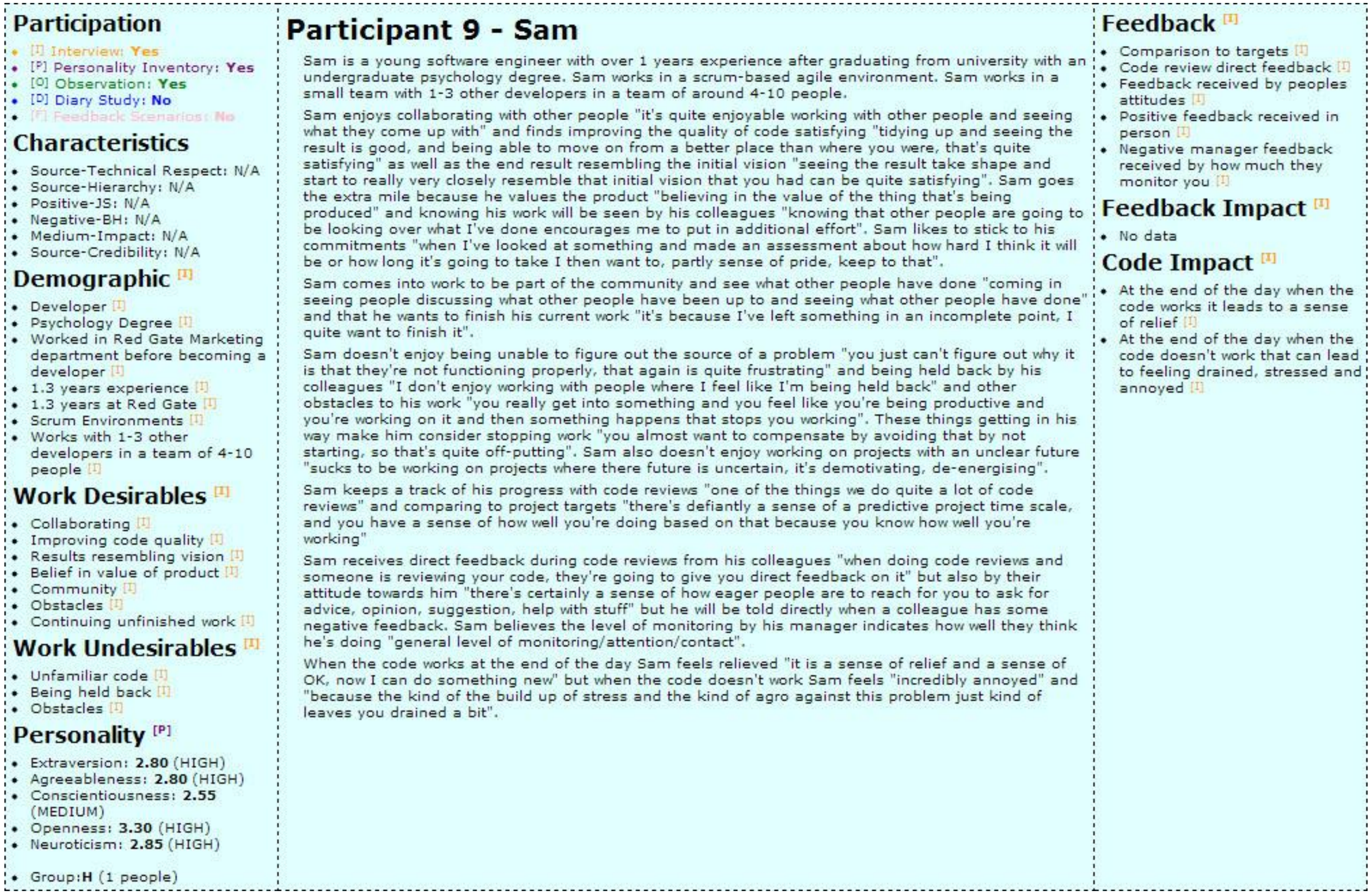
Participant 8 - Cory

Greg

Feedback Impact [F]

- Positive face to face or emailed peer feedback has an impact on job satisfaction. [F]
- Negative project manager or division head progress feedback during a stand up meeting has an impact on attitude, motivation, productivity and job satisfaction. [F]
- Positive division head performance emailed feedback has an impact on attitude, motivation and job satisfaction. The same feedback from the project manager would have less impact as they are "a lot closer to you" and "recognition from senior management is always a good thing". The same feedback from the CEO "means a bit more" because they're "further away from you". [F]
- Negative peer feedback relayed through the project manager has an impact on behaviour. The same feedback relayed from the division head would have a varied impact based on the content of the feedback. [F]
- Positive indirect feedback from a peer or project manager has an impact on motivation and job satisfaction. [F]
- Negative face to face or emailed peer feedback on breaking the build has an impact on behaviour. [F]
- Positive delayed performance feedback has an impact on motivation and job satisfaction. [F]

PARTICIPANT 8 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **No**
- [F] Feedback Scenarios: **No**

Characteristics

- Source-Technical Respect: N/A
- Source-Hierarchy: N/A
- Positive-JS: N/A
- Negative-BH: N/A
- Medium-Impact: N/A
- Source-Credibility: N/A

Demographic [I]

- Developer [I]
- Psychology Degree [I]
- Worked in Red Gate Marketing department before becoming a developer [I]
- 1.3 years experience [I]
- 1.3 years at Red Gate [I]
- Scrum Environments [I]
- Works with 1-3 other developers in a team of 4-10 people [I]

Work Desirables [I]

- Collaborating [I]
- Improving code quality [I]
- Results resembling vision [I]
- Belief in value of product [I]
- Community [I]
- Obstacles [I]
- Continuing unfinished work [I]

Work Undesirables [I]

- Unfamiliar code [I]
- Being held back [I]
- Obstacles [I]

Personality [P]

- Extraversion: **2.80** (HIGH)
- Agreeableness: **2.80** (HIGH)
- Conscientiousness: **2.55** (MEDIUM)
- Openness: **3.30** (HIGH)
- Neuroticism: **2.85** (HIGH)

- Group:**H** (1 people)

Participant 9 - Sam

Sam is a young software engineer with over 1 years experience after graduating from university with an undergraduate psychology degree. Sam works in a scrum-based agile environment. Sam works in a small team with 1-3 other developers in a team of around 4-10 people.

Sam enjoys collaborating with other people "it's quite enjoyable working with other people and seeing what they come up with" and finds improving the quality of code satisfying "tidying up and seeing the result is good, and being able to move on from a better place than where you were, that's quite satisfying" as well as the end result resembling the initial vision "seeing the result take shape and start to really very closely resemble that initial vision that you had can be quite satisfying". Sam goes the extra mile because he values the product "believing in the value of the thing that's being produced" and knowing his work will be seen by his colleagues "knowing that other people are going to be looking over what I've done encourages me to put in additional effort". Sam likes to stick to his commitments "when I've looked at something and made an assessment about how hard I think it will be or how long it's going to take I then want to, partly sense of pride, keep to that".

Sam comes into work to be part of the community and see what other people have done "coming in seeing people discussing what other people have been up to and seeing what other people have done" and that he wants to finish his current work "it's because I've left something in an incomplete point, I quite want to finish it".

Sam doesn't enjoy being unable to figure out the source of a problem "you just can't figure out why it is that they're not functioning properly, that again is quite frustrating" and being held back by his colleagues "I don't enjoy working with people where I feel like I'm being held back" and other obstacles to his work "you really get into something and you feel like you're being productive and you're working on it and then something happens that stops you working". These things getting in his way make him consider stopping work "you almost want to compensate by avoiding that by not starting, so that's quite off-putting". Sam also doesn't enjoy working on projects with an unclear future "sucks to be working on projects where there future is uncertain, it's demotivating, de-energising".

Sam keeps a track of his progress with code reviews "one of the things we do quite a lot of code reviews" and comparing to project targets "there's defiantly a sense of a predictive project time scale, and you have a sense of how well you're doing based on that because you know how well you're working"

Sam receives direct feedback during code reviews from his colleagues "when doing code reviews and someone is reviewing your code, they're going to give you direct feedback on it" but also by their attitude towards him "there's certainly a sense of how eager people are to reach for you to ask for advice, opinion, suggestion, help with stuff" but he will be told directly when a colleague has some negative feedback. Sam believes the level of monitoring by his manager indicates how well they think he's doing "general level of monitoring/attention/contact".

When the code works at the end of the day Sam feels relieved "it is a sense of relief and a sense of OK, now I can do something new" but when the code doesn't work Sam feels "incredibly annoyed" and "because the kind of the build up of stress and the kind of agro against this problem just kind of leaves you drained a bit".

Feedback [I]

- Comparison to targets [I]
- Code review direct feedback [I]
- Feedback received by peoples attitudes [I]
- Positive feedback received in person [I]
- Negative manager feedback received by how much they monitor you [I]

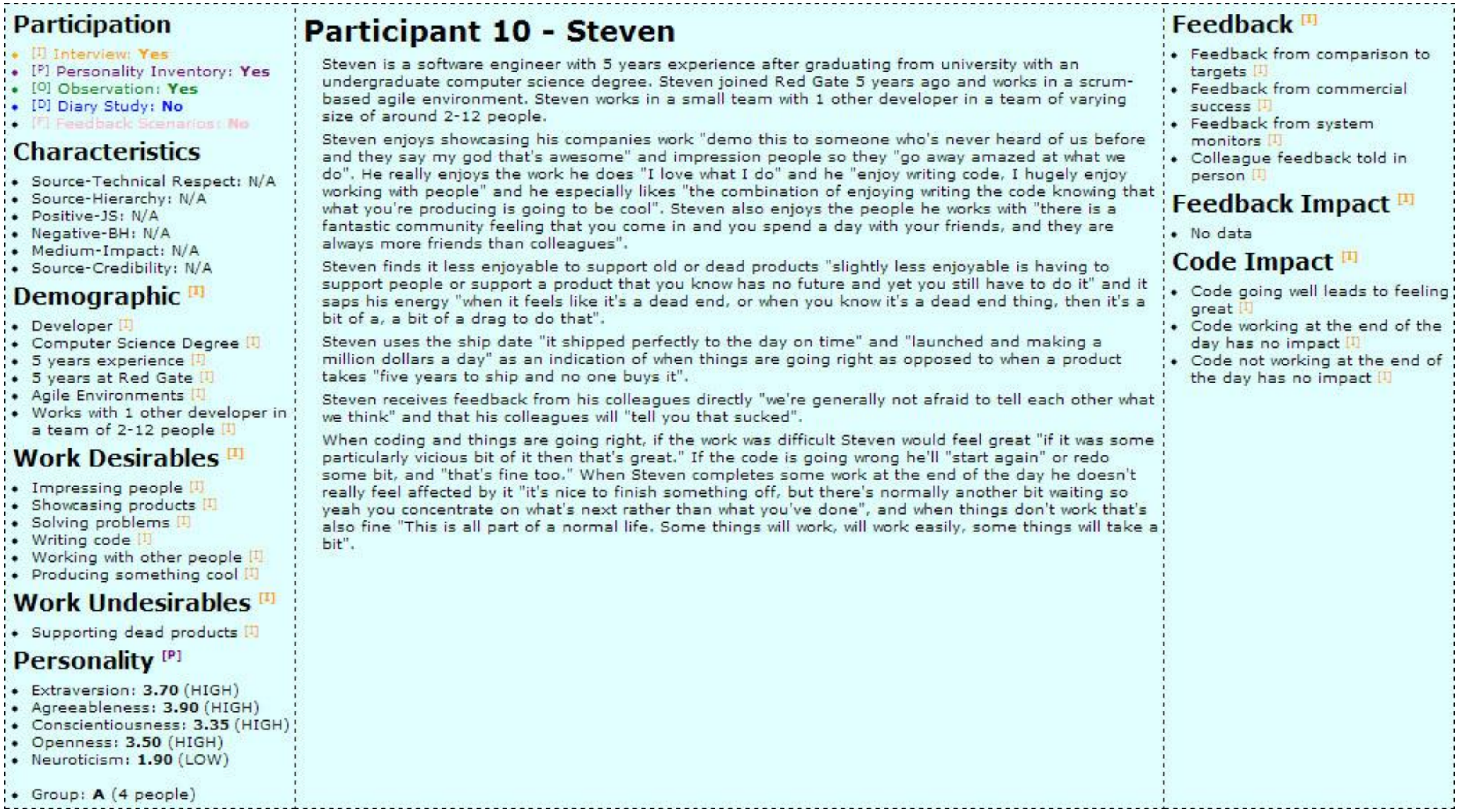
Feedback Impact [I]

- No data

Code Impact [I]

- At the end of the day when the code works it leads to a sense of relief [I]
- At the end of the day when the code doesn't work that can lead to feeling drained, stressed and annoyed [I]

PARTICIPANT 9 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **No**
- [F] Feedback Scenarios: **No**

Characteristics

- Source-Technical Respect: N/A
- Source-Hierarchy: N/A
- Positive-JS: N/A
- Negative-BH: N/A
- Medium-Impact: N/A
- Source-Credibility: N/A

Demographic [I]

- Developer [I]
- Computer Science Degree [I]
- 5 years experience [I]
- 5 years at Red Gate [I]
- Agile Environments [I]
- Works with 1 other developer in a team of 2-12 people [I]

Work Desirables [I]

- Impressing people [I]
- Showcasing products [I]
- Solving problems [I]
- Writing code [I]
- Working with other people [I]
- Producing something cool [I]

Work Undesirables [I]

- Supporting dead products [I]

Personality [P]

- Extraversion: **3.70** (HIGH)
- Agreeableness: **3.90** (HIGH)
- Conscientiousness: **3.35** (HIGH)
- Openness: **3.50** (HIGH)
- Neuroticism: **1.90** (LOW)

- Group: **A** (4 people)

Participant 10 - Steven

Steven is a software engineer with 5 years experience after graduating from university with an undergraduate computer science degree. Steven joined Red Gate 5 years ago and works in a scrum-based agile environment. Steven works in a small team with 1 other developer in a team of varying size of around 2-12 people.

Steven enjoys showcasing his companies work "demo this to someone who's never heard of us before and they say my god that's awesome" and impression people so they "go away amazed at what we do". He really enjoys the work he does "I love what I do" and he "enjoy writing code, I hugely enjoy working with people" and he especially likes "the combination of enjoying writing the code knowing that what you're producing is going to be cool". Steven also enjoys the people he works with "there is a fantastic community feeling that you come in and you spend a day with your friends, and they are always more friends than colleagues".

Steven finds it less enjoyable to support old or dead products "slightly less enjoyable is having to support people or support a product that you know has no future and yet you still have to do it" and it saps his energy "when it feels like it's a dead end, or when you know it's a dead end thing, then it's a bit of a, a bit of a drag to do that".

Steven uses the ship date "it shipped perfectly to the day on time" and "launched and making a million dollars a day" as an indication of when things are going right as opposed to when a product takes "five years to ship and no one buys it".

Steven receives feedback from his colleagues directly "we're generally not afraid to tell each other what we think" and that his colleagues will "tell you that sucked".

When coding and things are going right, if the work was difficult Steven would feel great "if it was some particularly vicious bit of it then that's great." If the code is going wrong he'll "start again" or redo some bit, and "that's fine too." When Steven completes some work at the end of the day he doesn't really feel affected by it "it's nice to finish something off, but there's normally another bit waiting so yeah you concentrate on what's next rather than what you've done", and when things don't work that's also fine "This is all part of a normal life. Some things will work, will work easily, some things will take a bit".

Feedback [I]

- Feedback from comparison to targets [I]
- Feedback from commercial success [I]
- Feedback from system monitors [I]
- Colleague feedback told in person [I]

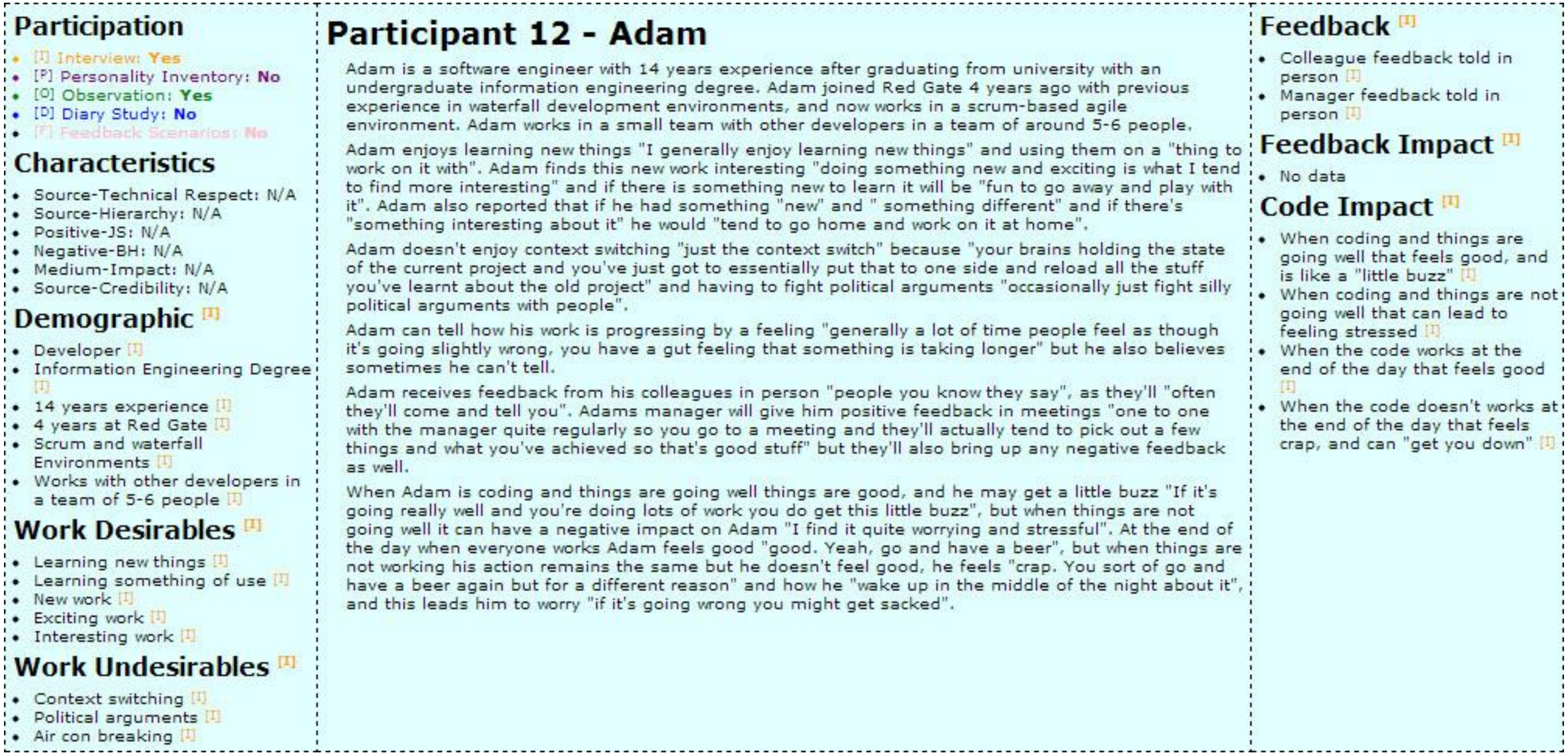
Feedback Impact [I]

- No data

Code Impact [I]

- Code going well leads to feeling great [I]
- Code working at the end of the day has no impact [I]
- Code not working at the end of the day has no impact [I]

PARTICIPANT 10 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **No**
- [O] Observation: **Yes**
- [D] Diary Study: **No**
- [F] Feedback Scenarios: **No**

Characteristics

- Source-Technical Respect: N/A
- Source-Hierarchy: N/A
- Positive-JS: N/A
- Negative-BH: N/A
- Medium-Impact: N/A
- Source-Credibility: N/A

Demographic [I]

- Developer [I]
- Information Engineering Degree [I]
- 14 years experience [I]
- 4 years at Red Gate [I]
- Scrum and waterfall Environments [I]
- Works with other developers in a team of 5-6 people [I]

Work Desirables [I]

- Learning new things [I]
- Learning something of use [I]
- New work [I]
- Exciting work [I]
- Interesting work [I]

Work Undesirables [I]

- Context switching [I]
- Political arguments [I]
- Air con breaking [I]

Participant 12 - Adam

Adam is a software engineer with 14 years experience after graduating from university with an undergraduate information engineering degree. Adam joined Red Gate 4 years ago with previous experience in waterfall development environments, and now works in a scrum-based agile environment. Adam works in a small team with other developers in a team of around 5-6 people.

Adam enjoys learning new things "I generally enjoy learning new things" and using them on a "thing to work on it with". Adam finds this new work interesting "doing something new and exciting is what I tend to find more interesting" and if there is something new to learn it will be "fun to go away and play with it". Adam also reported that if he had something "new" and " something different" and if there's "something interesting about it" he would "tend to go home and work on it at home".

Adam doesn't enjoy context switching "just the context switch" because "your brains holding the state of the current project and you've just got to essentially put that to one side and reload all the stuff you've learnt about the old project" and having to fight political arguments "occasionally just fight silly political arguments with people".

Adam can tell how his work is progressing by a feeling "generally a lot of time people feel as though it's going slightly wrong, you have a gut feeling that something is taking longer" but he also believes sometimes he can't tell.

Adam receives feedback from his colleagues in person "people you know they say", as they'll "often they'll come and tell you". Adams manager will give him positive feedback in meetings "one to one with the manager quite regularly so you go to a meeting and they'll actually tend to pick out a few things and what you've achieved so that's good stuff" but they'll also bring up any negative feedback as well.

When Adam is coding and things are going well things are good, and he may get a little buzz "If it's going really well and you're doing lots of work you do get this little buzz", but when things are not going well it can have a negative impact on Adam "I find it quite worrying and stressful". At the end of the day when everyone works Adam feels good "good. Yeah, go and have a beer", but when things are not working his action remains the same but he doesn't feel good, he feels "crap. You sort of go and have a beer again but for a different reason" and how he "wake up in the middle of the night about it", and this leads him to worry "if it's going wrong you might get sacked".

Feedback [I]

- Colleague feedback told in person [I]
- Manager feedback told in person [I]

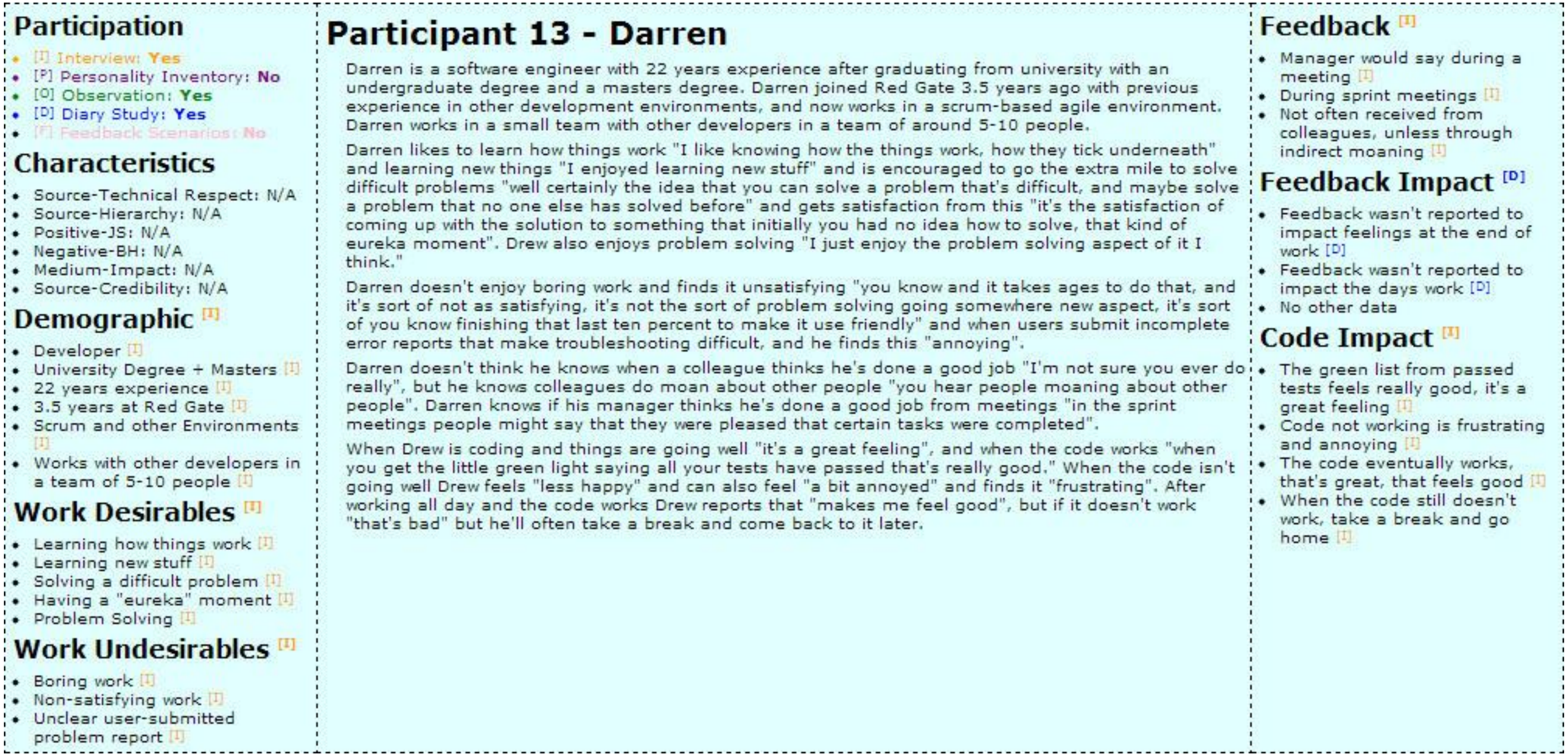
Feedback Impact [I]

- No data

Code Impact [I]

- When coding and things are going well that feels good, and is like a "little buzz" [I]
- When coding and things are not going well that can lead to feeling stressed [I]
- When the code works at the end of the day that feels good [I]
- When the code doesn't works at the end of the day that feels crap, and can "get you down" [I]

PARTICIPANT 12 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **No**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **No**

Characteristics

- Source-Technical Respect: N/A
- Source-Hierarchy: N/A
- Positive-JS: N/A
- Negative-BH: N/A
- Medium-Impact: N/A
- Source-Credibility: N/A

Demographic [I]

- Developer [I]
- University Degree + Masters [I]
- 22 years experience [I]
- 3.5 years at Red Gate [I]
- Scrum and other Environments [I]
- Works with other developers in a team of 5-10 people [I]

Work Desirables [I]

- Learning how things work [I]
- Learning new stuff [I]
- Solving a difficult problem [I]
- Having a "eureka" moment [I]
- Problem Solving [I]

Work Undesirables [I]

- Boring work [I]
- Non-satisfying work [I]
- Unclear user-submitted problem report [I]

Participant 13 - Darren

Darren is a software engineer with 22 years experience after graduating from university with an undergraduate degree and a masters degree. Darren joined Red Gate 3.5 years ago with previous experience in other development environments, and now works in a scrum-based agile environment. Darren works in a small team with other developers in a team of around 5-10 people.

Darren likes to learn how things work "I like knowing how the things work, how they tick underneath" and learning new things "I enjoyed learning new stuff" and is encouraged to go the extra mile to solve difficult problems "well certainly the idea that you can solve a problem that's difficult, and maybe solve a problem that no one else has solved before" and gets satisfaction from this "it's the satisfaction of coming up with the solution to something that initially you had no idea how to solve, that kind of eureka moment". Drew also enjoys problem solving "I just enjoy the problem solving aspect of it I think."

Darren doesn't enjoy boring work and finds it unsatisfying "you know and it takes ages to do that, and it's sort of not as satisfying, it's not the sort of problem solving going somewhere new aspect, it's sort of you know finishing that last ten percent to make it use friendly" and when users submit incomplete error reports that make troubleshooting difficult, and he finds this "annoying".

Darren doesn't think he knows when a colleague thinks he's done a good job "I'm not sure you ever do really", but he knows colleagues do moan about other people "you hear people moaning about other people". Darren knows if his manager thinks he's done a good job from meetings "in the sprint meetings people might say that they were pleased that certain tasks were completed".

When Drew is coding and things are going well "it's a great feeling", and when the code works "when you get the little green light saying all your tests have passed that's really good." When the code isn't going well Drew feels "less happy" and can also feel "a bit annoyed" and finds it "frustrating". After working all day and the code works Drew reports that "makes me feel good", but if it doesn't work "that's bad" but he'll often take a break and come back to it later.

Feedback [I]

- Manager would say during a meeting [I]
- During sprint meetings [I]
- Not often received from colleagues, unless through indirect moaning [I]

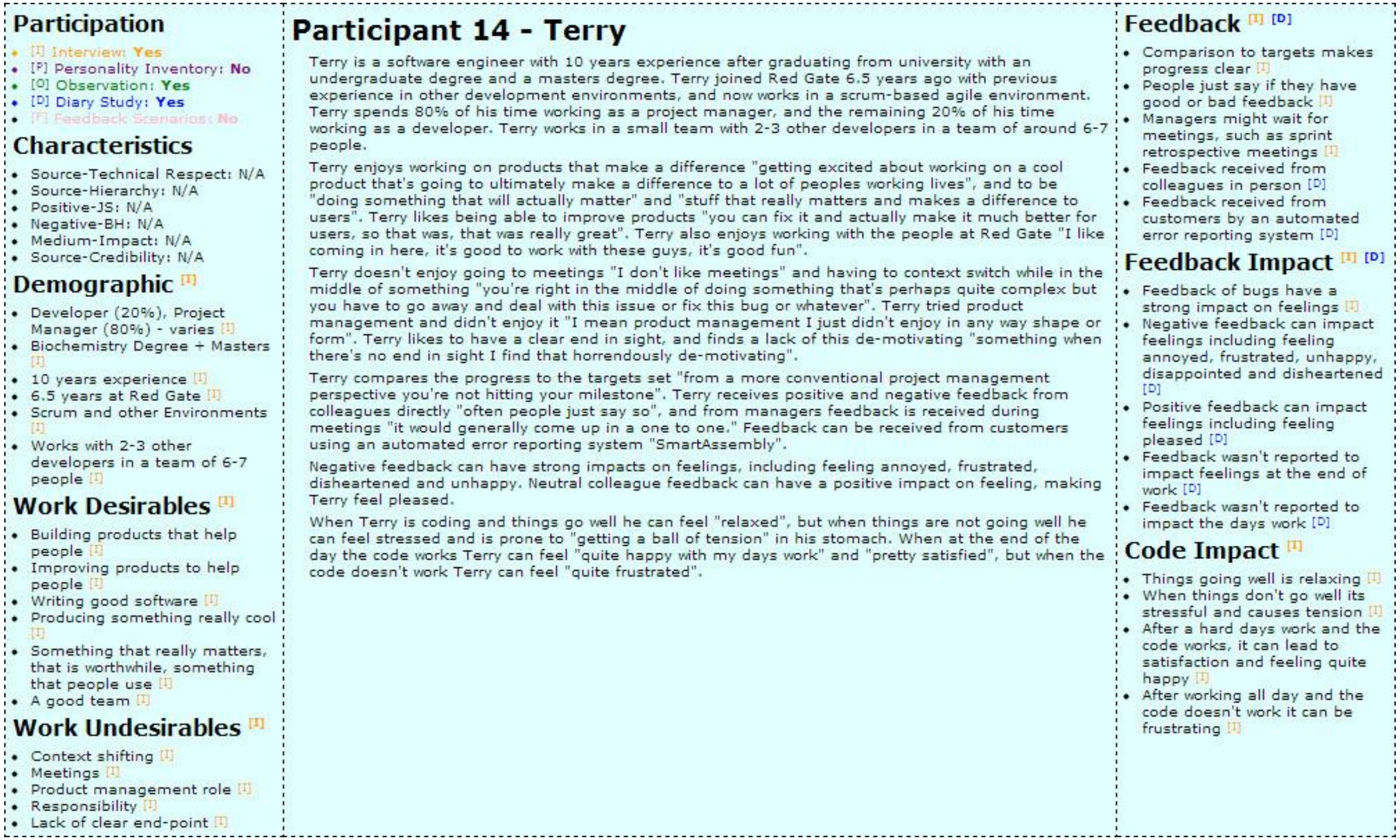
Feedback Impact [D]

- Feedback wasn't reported to impact feelings at the end of work [D]
- Feedback wasn't reported to impact the days work [D]
- No other data

Code Impact [I]

- The green list from passed tests feels really good, it's a great feeling [I]
- Code not working is frustrating and annoying [I]
- The code eventually works, that's great, that feels good [I]
- When the code still doesn't work, take a break and go home [I]

PARTICIPANT 13 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **No**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **No**

Characteristics

- Source-Technical Respect: N/A
- Source-Hierarchy: N/A
- Positive-JS: N/A
- Negative-BH: N/A
- Medium-Impact: N/A
- Source-Credibility: N/A

Demographic [I]

- Developer (20%), Project Manager (80%) - varies [I]
- Biochemistry Degree + Masters [I]
- 10 years experience [I]
- 6.5 years at Red Gate [I]
- Scrum and other Environments [I]
- Works with 2-3 other developers in a team of 6-7 people [I]

Work Desirables [I]

- Building products that help people [I]
- Improving products to help people [I]
- Writing good software [I]
- Producing something really cool [I]
- Something that really matters, that is worthwhile, something that people use [I]
- A good team [I]

Work Undesirables [I]

- Context shifting [I]
- Meetings [I]
- Product management role [I]
- Responsibility [I]
- Lack of clear end-point [I]

Participant 14 - Terry

Terry is a software engineer with 10 years experience after graduating from university with an undergraduate degree and a masters degree. Terry joined Red Gate 6.5 years ago with previous experience in other development environments, and now works in a scrum-based agile environment. Terry spends 80% of his time working as a project manager, and the remaining 20% of his time working as a developer. Terry works in a small team with 2-3 other developers in a team of around 6-7 people.

Terry enjoys working on products that make a difference "getting excited about working on a cool product that's going to ultimately make a difference to a lot of peoples working lives", and to be "doing something that will actually matter" and "stuff that really matters and makes a difference to users". Terry likes being able to improve products "you can fix it and actually make it much better for users, so that was, that was really great". Terry also enjoys working with the people at Red Gate "I like coming in here, it's good to work with these guys, it's good fun".

Terry doesn't enjoy going to meetings "I don't like meetings" and having to context switch while in the middle of something "you're right in the middle of doing something that's perhaps quite complex but you have to go away and deal with this issue or fix this bug or whatever". Terry tried product management and didn't enjoy it "I mean product management I just didn't enjoy in any way shape or form". Terry likes to have a clear end in sight, and finds a lack of this de-motivating "something when there's no end in sight I find that horrendously de-motivating".

Terry compares the progress to the targets set "from a more conventional project management perspective you're not hitting your milestone". Terry receives positive and negative feedback from colleagues directly "often people just say so", and from managers feedback is received during meetings "it would generally come up in a one to one." Feedback can be received from customers using an automated error reporting system "SmartAssembly".

Negative feedback can have strong impacts on feelings, including feeling annoyed, frustrated, disheartened and unhappy. Neutral colleague feedback can have a positive impact on feeling, making Terry feel pleased.

When Terry is coding and things go well he can feel "relaxed", but when things are not going well he can feel stressed and is prone to "getting a ball of tension" in his stomach. When at the end of the day the code works Terry can feel "quite happy with my days work" and "pretty satisfied", but when the code doesn't work Terry can feel "quite frustrated".

Feedback [I] [D]

- Comparison to targets makes progress clear [I]
- People just say if they have good or bad feedback [I]
- Managers might wait for meetings, such as sprint retrospective meetings [I]
- Feedback received from colleagues in person [D]
- Feedback received from customers by an automated error reporting system [D]

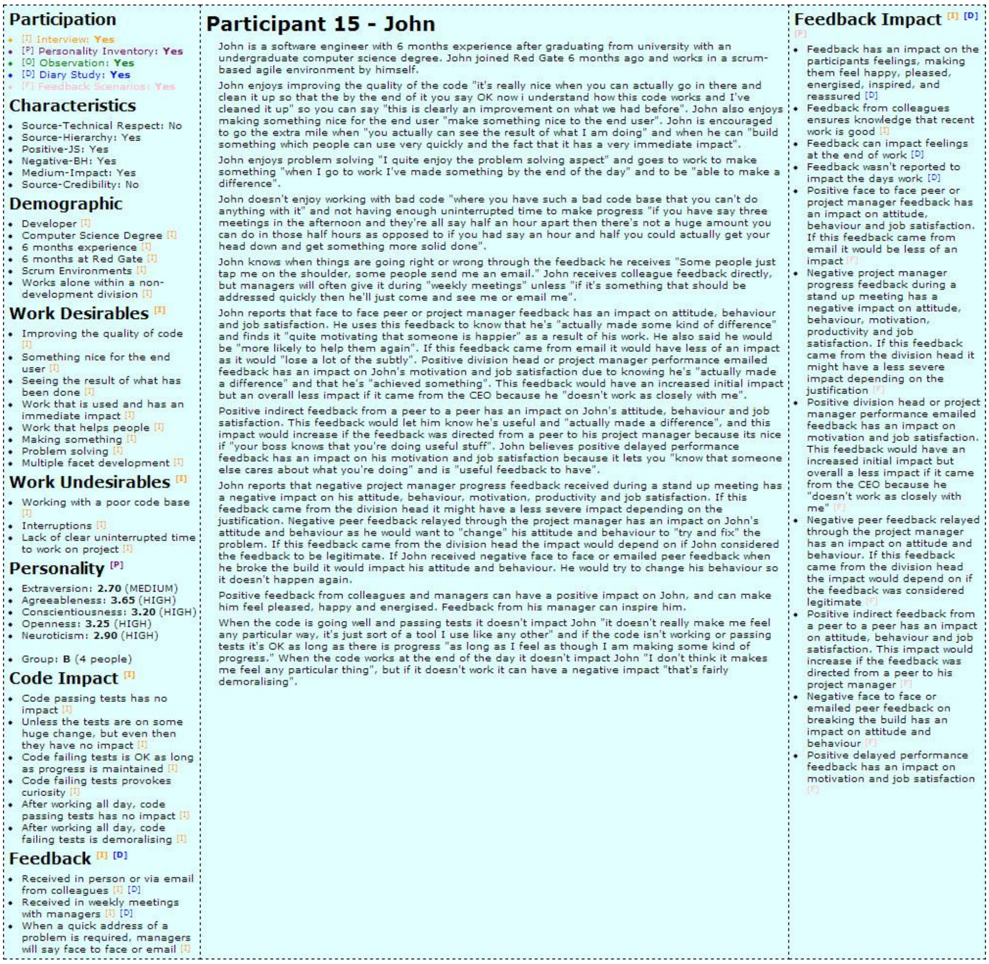
Feedback Impact [I] [D]

- Feedback of bugs have a strong impact on feelings [I]
- Negative feedback can impact feelings including feeling annoyed, frustrated, unhappy, disappointed and disheartened [D]
- Positive feedback can impact feelings including feeling pleased [D]
- Feedback wasn't reported to impact feelings at the end of work [D]
- Feedback wasn't reported to impact the days work [D]

Code Impact [I]

- Things going well is relaxing [I]
- When things don't go well its stressful and causes tension [I]
- After a hard days work and the code works, it can lead to satisfaction and feeling quite happy [I]
- After working all day and the code doesn't work it can be frustrating [I]

PARTICIPANT 14 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **Yes**

Characteristics

- Source-Technical Respect: No
- Source-Hierarchy: Yes
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: Yes
- Source-Credibility: No

Demographic

- Developer [I]
- Computer Science Degree [I]
- 6 months experience [I]
- 6 months at Red Gate [I]
- Scrum Environments [I]
- Works alone within a non-development division [I]

Work Desirables [I]

- Improving the quality of code [I]
- Something nice for the end user [I]
- Seeing the result of what has been done [I]
- Work that is used and has an immediate impact [I]
- Work that helps people [I]
- Making something [I]
- Problem solving [I]
- Multiple facet development [I]

Work Undesirables [I]

- Working with a poor code base [I]
- Interruptions [I]
- Lack of clear uninterrupted time to work on project [I]

Personality [P]

- Extraversion: **2.70** (MEDIUM)
- Agreeableness: **3.65** (HIGH)
- Conscientiousness: **3.20** (HIGH)
- Openness: **3.25** (HIGH)
- Neuroticism: **2.90** (HIGH)

- Group: **B** (4 people)

Code Impact [I]

- Code passing tests has no impact [I]
- Unless the tests are on some huge change, but even then they have no impact [I]
- Code failing tests is OK as long as progress is maintained [I]
- Code failing tests provokes curiosity [I]
- After working all day, code passing tests has no impact [I]
- After working all day, code failing tests is demoralising [I]

Feedback [I] [D]

- Received in person or via email from colleagues [I] [D]
- Received in weekly meetings with managers [I] [D]
- When a quick address of a problem is required, managers will say face to face or email [I]

Participant 15 - John

John is a software engineer with 6 months experience after graduating from university with an undergraduate computer science degree. John joined Red Gate 6 months ago and works in a scrum-based agile environment by himself.

John enjoys improving the quality of the code "it's really nice when you can actually go in there and clean it up so that the by the end of it you say OK now i understand how this code works and I've cleaned it up" so you can say "this is clearly an improvement on what we had before". John also enjoys making something nice for the end user "make something nice to the end user". John is encouraged to go the extra mile when "you actually can see the result of what I am doing" and when he can "build something which people can use very quickly and the fact that it has a very immediate impact".

John enjoys problem solving "I quite enjoy the problem solving aspect" and goes to work to make something "when I go to work I've made something by the end of the day" and to be "able to make a difference".

John doesn't enjoy working with bad code "where you have such a bad code base that you can't do anything with it" and not having enough uninterrupted time to make progress "if you have say three meetings in the afternoon and they're all say half an hour apart then there's not a huge amount you can do in those half hours as opposed to if you had say an hour and half you could actually get your head down and get something more solid done".

John knows when things are going right or wrong through the feedback he receives "Some people just tap me on the shoulder, some people send me an email." John receives colleague feedback directly, but managers will often give it during "weekly meetings" unless "if it's something that should be addressed quickly then he'll just come and see me or email me".

John reports that face to face peer or project manager feedback has an impact on attitude, behaviour and job satisfaction. He uses this feedback to know that he's "actually made some kind of difference" and finds it "quite motivating that someone is happier" as a result of his work. He also said he would be "more likely to help them again". If this feedback came from email it would have less of an impact as it would "lose a lot of the subtly". Positive division head or project manager performance emailed feedback has an impact on John's motivation and job satisfaction due to knowing he's "actually made a difference" and that he's "achieved something". This feedback would have an increased initial impact but an overall less impact if it came from the CEO because he "doesn't work as closely with me".

Positive indirect feedback from a peer to a peer has an impact on John's attitude, behaviour and job satisfaction. This feedback would let him know he's useful and "actually made a difference", and this impact would increase if the feedback was directed from a peer to his project manager because its nice if "your boss knows that you're doing useful stuff". John believes positive delayed performance feedback has an impact on his motivation and job satisfaction because it lets you "know that someone else cares about what you're doing" and is "useful feedback to have".

John reports that negative project manager progress feedback received during a stand up meeting has a negative impact on his attitude, behaviour, motivation, productivity and job satisfaction. If this feedback came from the division head it might have a less severe impact depending on the justification. Negative peer feedback relayed through the project manager has an impact on John's attitude and behaviour as he would want to "change" his attitude and behaviour to "try and fix" the problem. If this feedback came from the division head the impact would depend on if John considered the feedback to be legitimate. If John received negative face to face or emailed peer feedback when he broke the build it would impact his attitude and behaviour. He would try to change his behaviour so it doesn't happen again.

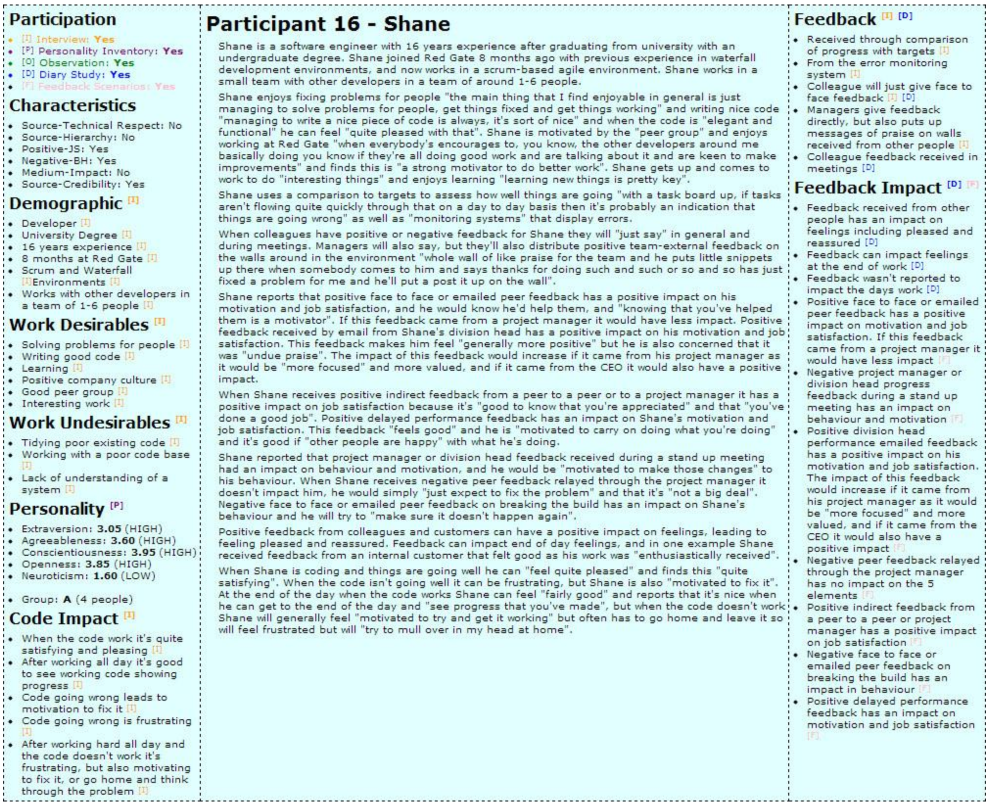
Positive feedback from colleagues and managers can have a positive impact on John, and can make him feel pleased, happy and energised. Feedback from his manager can inspire him.

When the code is going well and passing tests it doesn't impact John "it doesn't really make me feel any particular way, it's just sort of a tool I use like any other" and if the code isn't working or passing tests it's OK as long as there is progress "as long as I feel as though I am making some kind of progress." When the code works at the end of the day it doesn't impact John "I don't think it makes me feel any particular thing", but if it doesn't work it can have a negative impact "that's fairly demoralising".

Feedback Impact [I] [D] [F]

- Feedback has an impact on the participants feelings, making them feel happy, pleased, energised, inspired, and reassured [D]
- Feedback from colleagues ensures knowledge that recent work is good [I]
- Feedback can impact feelings at the end of work [D]
- Feedback wasn't reported to impact the days work [D]
- Positive face to face peer or project manager feedback has an impact on attitude, behaviour and job satisfaction. If this feedback came from email it would be less of an impact [F]
- Negative project manager progress feedback during a stand up meeting has a negative impact on attitude, behaviour, motivation, productivity and job satisfaction. If this feedback came from the division head it might have a less severe impact depending on the justification [F]
- Positive division head or project manager performance emailed feedback has an impact on motivation and job satisfaction. This feedback would have an increased initial impact but overall a less impact if it came from the CEO because he "doesn't work as closely with me" [F]
- Negative peer feedback relayed through the project manager has an impact on attitude and behaviour. If this feedback came from the division head the impact would depend on if the feedback was considered legitimate [F]
- Positive indirect feedback from a peer to a peer has an impact on attitude, behaviour and job satisfaction. This impact would increase if the feedback was directed from a peer to his project manager [F]
- Negative face to face or emailed peer feedback on breaking the build has an impact on attitude and behaviour [F]
- Positive delayed performance feedback has an impact on motivation and job satisfaction [F]

PARTICIPANT 15 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **Yes**

Characteristics

- Source-Technical Respect: No
- Source-Hierarchy: No
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: No
- Source-Credibility: Yes

Demographic [I]

- Developer [I]
- University Degree [I]
- 16 years experience [I]
- 8 months at Red Gate [I]
- Scrum and Waterfall [I]Environments [I]
- Works with other developers in a team of 1-6 people [I]

Work Desirables [I]

- Solving problems for people [I]
- Writing good code [I]
- Learning [I]
- Positive company culture [I]
- Good peer group [I]
- Interesting work [I]

Work Undesirables [I]

- Tidying poor existing code [I]
- Working with a poor code base [I]
- Lack of understanding of a system [I]

Personality [P]

- Extraversion: **3.05** (HIGH)
- Agreeableness: **3.60** (HIGH)
- Conscientiousness: **3.95** (HIGH)
- Openness: **3.85** (HIGH)
- Neuroticism: **1.60** (LOW)

- Group: **A** (4 people)

Code Impact [I]

- When the code work it's quite satisfying and pleasing [I]
- After working all day it's good to see working code showing progress [I]
- Code going wrong leads to motivation to fix it [I]
- Code going wrong is frustrating [I]
- After working hard all day and the code doesn't work it's frustrating, but also motivating to fix it, or go home and think through the problem [I]

Participant 16 - Shane

Shane is a software engineer with 16 years experience after graduating from university with an undergraduate degree. Shane joined Red Gate 8 months ago with previous experience in waterfall development environments, and now works in a scrum-based agile environment. Shane works in a small team with other developers in a team of around 1-6 people.

Shane enjoys fixing problems for people "the main thing that I find enjoyable in general is just managing to solve problems for people, get things fixed and get things working" and writing nice code "managing to write a nice piece of code is always, it's sort of nice" and when the code is "elegant and functional" he can feel "quite pleased with that". Shane is motivated by the "peer group" and enjoys working at Red Gate "when everybody's encourages to, you know, the other developers around me basically doing you know if they're all doing good work and are talking about it and are keen to make improvements" and finds this is "a strong motivator to do better work". Shane gets up and comes to work to do "interesting things" and enjoys learning "learning new things is pretty key".

Shane uses a comparison to targets to assess how well things are going "with a task board up, if tasks aren't flowing quite quickly through that on a day to day basis then it's probably an indication that things are going wrong" as well as "monitoring systems" that display errors.

When colleagues have positive or negative feedback for Shane they will "just say" in general and during meetings. Managers will also say, but they'll also distribute positive team-external feedback on the walls around in the environment "whole wall of like praise for the team and he puts little snippets up there when somebody comes to him and says thanks for doing such and such or so and so has just fixed a problem for me and he'll put a post it up on the wall".

Shane reports that positive face to face or emailed peer feedback has a positive impact on his motivation and job satisfaction, and he would know he'd help them, and "knowing that you've helped them is a motivator". If this feedback came from a project manager it would have less impact. Positive feedback received by email from Shane's division head has a positive impact on his motivation and job satisfaction. This feedback makes him feel "generally more positive" but he is also concerned that it was "undue praise". The impact of this feedback would increase if it came from his project manager as it would be "more focused" and more valued, and if it came from the CEO it would also have a positive impact.

When Shane receives positive indirect feedback from a peer to a peer or to a project manager it has a positive impact on job satisfaction because it's "good to know that you're appreciated" and that "you've done a good job". Positive delayed performance feedback has an impact on Shane's motivation and job satisfaction. This feedback "feels good" and he is "motivated to carry on doing what you're doing" and it's good if "other people are happy" with what he's doing.

Shane reported that project manager or division head feedback received during a stand up meeting had an impact on behaviour and motivation, and he would be "motivated to make those changes" to his behaviour. When Shane receives negative peer feedback relayed through the project manager it doesn't impact him, he would simply "just expect to fix the problem" and that it's "not a big deal". Negative face to face or emailed peer feedback on breaking the build has an impact on Shane's behaviour and he will try to "make sure it doesn't happen again".

Positive feedback from colleagues and customers can have a positive impact on feelings, leading to feeling pleased and reassured. Feedback can impact end of day feelings, and in one example Shane received feedback from an internal customer that felt good as his work was "enthusiastically received".

When Shane is coding and things are going well he can "feel quite pleased" and finds this "quite satisfying". When the code isn't going well it can be frustrating, but Shane is also "motivated to fix it". At the end of the day when the code works Shane can feel "fairly good" and reports that it's nice when he can get to the end of the day and "see progress that you've made", but when the code doesn't work Shane will generally feel "motivated to try and get it working" but often has to go home and leave it so will feel frustrated but will "try to mull over in my head at home".

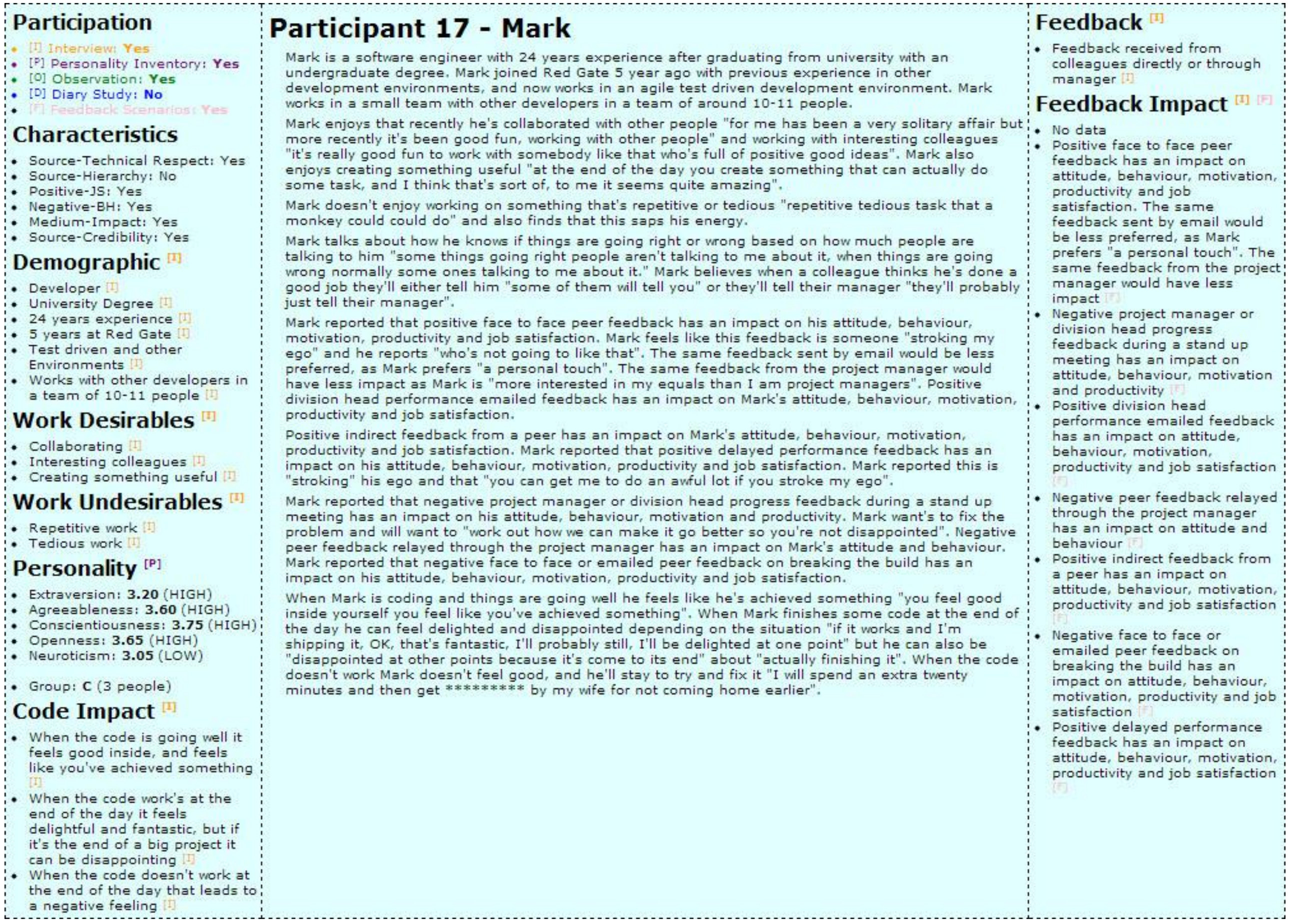
Feedback [I] [D]

- Received through comparison of progress with targets [I]
- From the error monitoring system [I]
- Colleague will just give face to face feedback [I] [D]
- Managers give feedback directly, but also puts up messages of praise on walls received from other people [I]
- Colleague feedback received in meetings [D]

Feedback Impact [D] [F]

- Feedback received from other people has an impact on feelings including pleased and reassured [D]
- Feedback can impact feelings at the end of work [D]
- Feedback wasn't reported to impact the days work [D]
- Positive face to face or emailed peer feedback has a positive impact on motivation and job satisfaction. If this feedback came from a project manager it would have less impact [F]
- Negative project manager or division head progress feedback during a stand up meeting has an impact on behaviour and motivation [F]
- Positive division head performance emailed feedback has a positive impact on his motivation and job satisfaction. The impact of this feedback would increase if it came from his project manager as it would be "more focused" and more valued, and if it came from the CEO it would also have a positive impact [F]
- Negative peer feedback relayed through the project manager has no impact on the 5 elements [F]
- Positive indirect feedback from a peer to a peer or project manager has a positive impact on job satisfaction [F]
- Negative face to face or emailed peer feedback on breaking the build has an impact in behaviour [F]
- Positive delayed performance feedback has an impact on motivation and job satisfaction [F]

PARTICIPANT 16 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **No**
- [F] Feedback Scenarios: **Yes**

Characteristics

- Source-Technical Respect: Yes
- Source-Hierarchy: No
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: Yes
- Source-Credibility: Yes

Demographic [I]

- Developer [I]
- University Degree [I]
- 24 years experience [I]
- 5 years at Red Gate [I]
- Test driven and other Environments [I]
- Works with other developers in a team of 10-11 people [I]

Work Desirables [I]

- Collaborating [I]
- Interesting colleagues [I]
- Creating something useful [I]

Work Undesirables [I]

- Repetitive work [I]
- Tedious work [I]

Personality [P]

- Extraversion: **3.20** (HIGH)
- Agreeableness: **3.60** (HIGH)
- Conscientiousness: **3.75** (HIGH)
- Openness: **3.65** (HIGH)
- Neuroticism: **3.05** (LOW)

- Group: **C** (3 people)

Code Impact [I]

- When the code is going well it feels good inside, and feels like you've achieved something [I]
- When the code work's at the end of the day it feels delightful and fantastic, but if it's the end of a big project it can be disappointing [I]
- When the code doesn't work at the end of the day that leads to a negative feeling [I]

Participant 17 - Mark

Mark is a software engineer with 24 years experience after graduating from university with an undergraduate degree. Mark joined Red Gate 5 year ago with previous experience in other development environments, and now works in an agile test driven development environment. Mark works in a small team with other developers in a team of around 10-11 people.

Mark enjoys that recently he's collaborated with other people "for me has been a very solitary affair but more recently it's been good fun, working with other people" and working with interesting colleagues "it's really good fun to work with somebody like that who's full of positive good ideas". Mark also enjoys creating something useful "at the end of the day you create something that can actually do some task, and I think that's sort of, to me it seems quite amazing".

Mark doesn't enjoy working on something that's repetitive or tedious "repetitive tedious task that a monkey could could do" and also finds that this saps his energy.

Mark talks about how he knows if things are going right or wrong based on how much people are talking to him "some things going right people aren't talking to me about it, when things are going wrong normally some ones talking to me about it." Mark believes when a colleague thinks he's done a good job they'll either tell him "some of them will tell you" or they'll tell their manager "they'll probably just tell their manager".

Mark reported that positive face to face peer feedback has an impact on his attitude, behaviour, motivation, productivity and job satisfaction. Mark feels like this feedback is someone "stroking my ego" and he reports "who's not going to like that". The same feedback sent by email would be less preferred, as Mark prefers "a personal touch". The same feedback from the project manager would have less impact as Mark is "more interested in my equals than I am project managers". Positive division head performance emailed feedback has an impact on Mark's attitude, behaviour, motivation, productivity and job satisfaction.

Positive indirect feedback from a peer has an impact on Mark's attitude, behaviour, motivation, productivity and job satisfaction. Mark reported that positive delayed performance feedback has an impact on his attitude, behaviour, motivation, productivity and job satisfaction. Mark reported this is "stroking" his ego and that "you can get me to do an awful lot if you stroke my ego".

Mark reported that negative project manager or division head progress feedback during a stand up meeting has an impact on his attitude, behaviour, motivation and productivity. Mark want's to fix the problem and will want to "work out how we can make it go better so you're not disappointed". Negative peer feedback relayed through the project manager has an impact on Mark's attitude and behaviour. Mark reported that negative face to face or emailed peer feedback on breaking the build has an impact on his attitude, behaviour, motivation, productivity and job satisfaction.

When Mark is coding and things are going well he feels like he's achieved something "you feel good inside yourself you feel like you've achieved something". When Mark finishes some code at the end of the day he can feel delighted and disappointed depending on the situation "if it works and I'm shipping it, OK, that's fantastic, I'll probably still, I'll be delighted at one point" but he can also be "disappointed at other points because it's come to its end" about "actually finishing it". When the code doesn't work Mark doesn't feel good, and he'll stay to try and fix it "I will spend an extra twenty minutes and then get ********* by my wife for not coming home earlier".

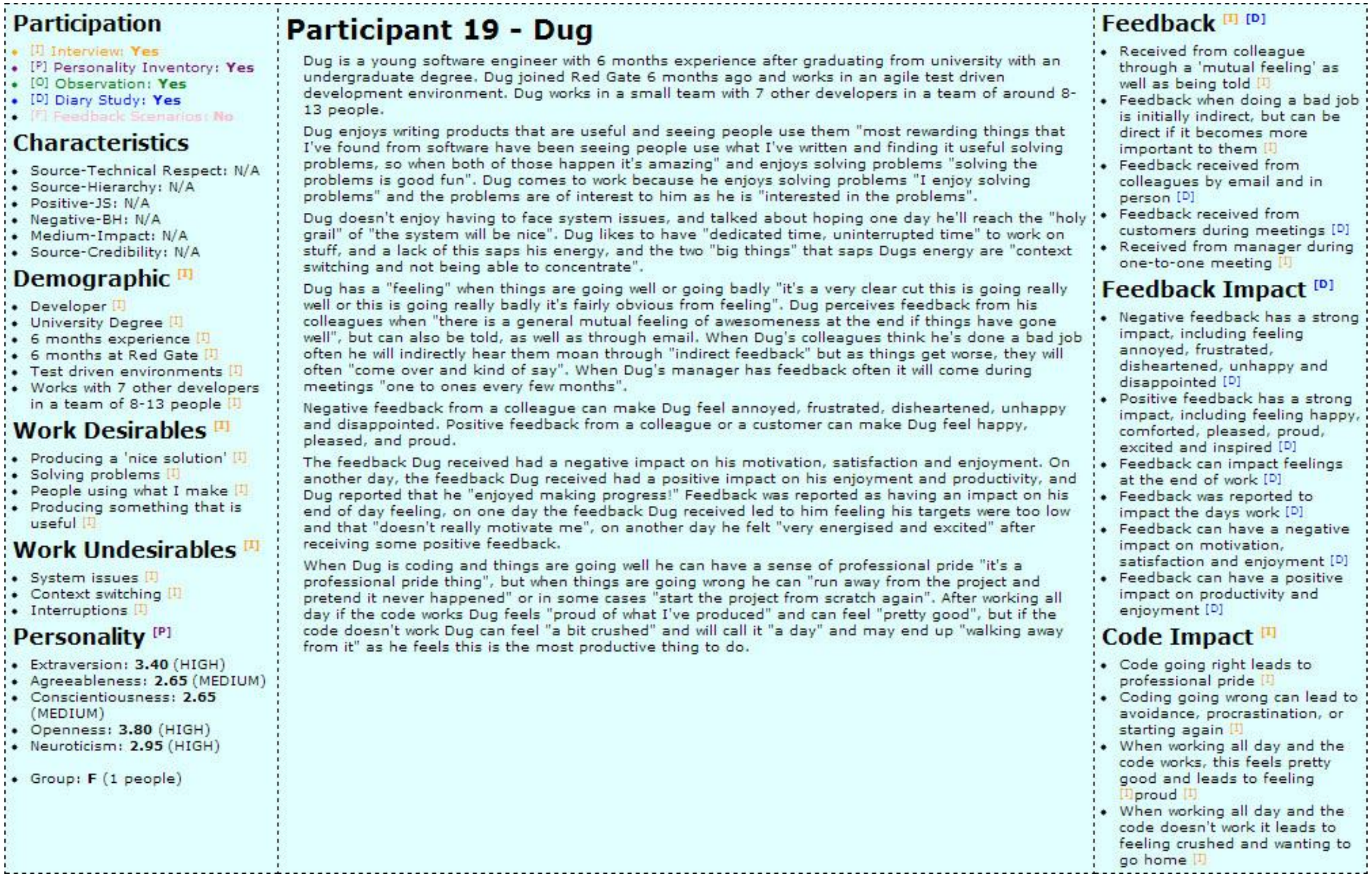
Feedback [I]

- Feedback received from colleagues directly or through manager [I]

Feedback Impact [I] [F]

- No data
- Positive face to face peer feedback has an impact on attitude, behaviour, motivation, productivity and job satisfaction. The same feedback sent by email would be less preferred, as Mark prefers "a personal touch". The same feedback from the project manager would have less impact [F]
- Negative project manager or division head progress feedback during a stand up meeting has an impact on attitude, behaviour, motivation and productivity [F]
- Positive division head performance emailed feedback has an impact on attitude, behaviour, motivation, productivity and job satisfaction [F]
- Negative peer feedback relayed through the project manager has an impact on attitude and behaviour [F]
- Positive indirect feedback from a peer has an impact on attitude, behaviour, motivation, productivity and job satisfaction [F]
- Negative face to face or emailed peer feedback on breaking the build has an impact on attitude, behaviour, motivation, productivity and job satisfaction [F]
- Positive delayed performance feedback has an impact on attitude, behaviour, motivation, productivity and job satisfaction [F]

PARTICIPANT 17 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **No**

Characteristics

- Source-Technical Respect: N/A
- Source-Hierarchy: N/A
- Positive-JS: N/A
- Negative-BH: N/A
- Medium-Impact: N/A
- Source-Credibility: N/A

Demographic [I]

- Developer [I]
- University Degree [I]
- 6 months experience [I]
- 6 months at Red Gate [I]
- Test driven environments [I]
- Works with 7 other developers in a team of 8-13 people [I]

Work Desirables [I]

- Producing a 'nice solution' [I]
- Solving problems [I]
- People using what I make [I]
- Producing something that is useful [I]

Work Undesirables [I]

- System issues [I]
- Context switching [I]
- Interruptions [I]

Personality [P]

- Extraversion: **3.40** (HIGH)
- Agreeableness: **2.65** (MEDIUM)
- Conscientiousness: **2.65** (MEDIUM)
- Openness: **3.80** (HIGH)
- Neuroticism: **2.95** (HIGH)

- Group: **F** (1 people)

Participant 19 - Dug

Dug is a young software engineer with 6 months experience after graduating from university with an undergraduate degree. Dug joined Red Gate 6 months ago and works in an agile test driven development environment. Dug works in a small team with 7 other developers in a team of around 8-13 people.

Dug enjoys writing products that are useful and seeing people use them "most rewarding things that I've found from software have been seeing people use what I've written and finding it useful solving problems, so when both of those happen it's amazing" and enjoys solving problems "solving the problems is good fun". Dug comes to work because he enjoys solving problems "I enjoy solving problems" and the problems are of interest to him as he is "interested in the problems".

Dug doesn't enjoy having to face system issues, and talked about hoping one day he'll reach the "holy grail" of "the system will be nice". Dug likes to have "dedicated time, uninterrupted time" to work on stuff, and a lack of this saps his energy, and the two "big things" that saps Dugs energy are "context switching and not being able to concentrate".

Dug has a "feeling" when things are going well or going badly "it's a very clear cut this is going really well or this is going really badly it's fairly obvious from feeling". Dug perceives feedback from his colleagues when "there is a general mutual feeling of awesomeness at the end if things have gone well", but can also be told, as well as through email. When Dug's colleagues think he's done a bad job often he will indirectly hear them moan through "indirect feedback" but as things get worse, they will often "come over and kind of say". When Dug's manager has feedback often it will come during meetings "one to ones every few months".

Negative feedback from a colleague can make Dug feel annoyed, frustrated, disheartened, unhappy and disappointed. Positive feedback from a colleague or a customer can make Dug feel happy, pleased, and proud.

The feedback Dug received had a negative impact on his motivation, satisfaction and enjoyment. On another day, the feedback Dug received had a positive impact on his enjoyment and productivity, and Dug reported that he "enjoyed making progress!" Feedback was reported as having an impact on his end of day feeling, on one day the feedback Dug received led to him feeling his targets were too low and that "doesn't really motivate me", on another day he felt "very energised and excited" after receiving some positive feedback.

When Dug is coding and things are going well he can have a sense of professional pride "it's a professional pride thing", but when things are going wrong he can "run away from the project and pretend it never happened" or in some cases "start the project from scratch again". After working all day if the code works Dug feels "proud of what I've produced" and can feel "pretty good", but if the code doesn't work Dug can feel "a bit crushed" and will call it "a day" and may end up "walking away from it" as he feels this is the most productive thing to do.

Feedback [I] [D]

- Received from colleague through a 'mutual feeling' as well as being told [I]
- Feedback when doing a bad job is initially indirect, but can be direct if it becomes more important to them [I]
- Feedback received from colleagues by email and in person [D]
- Feedback received from customers during meetings [D]
- Received from manager during one-to-one meeting [I]

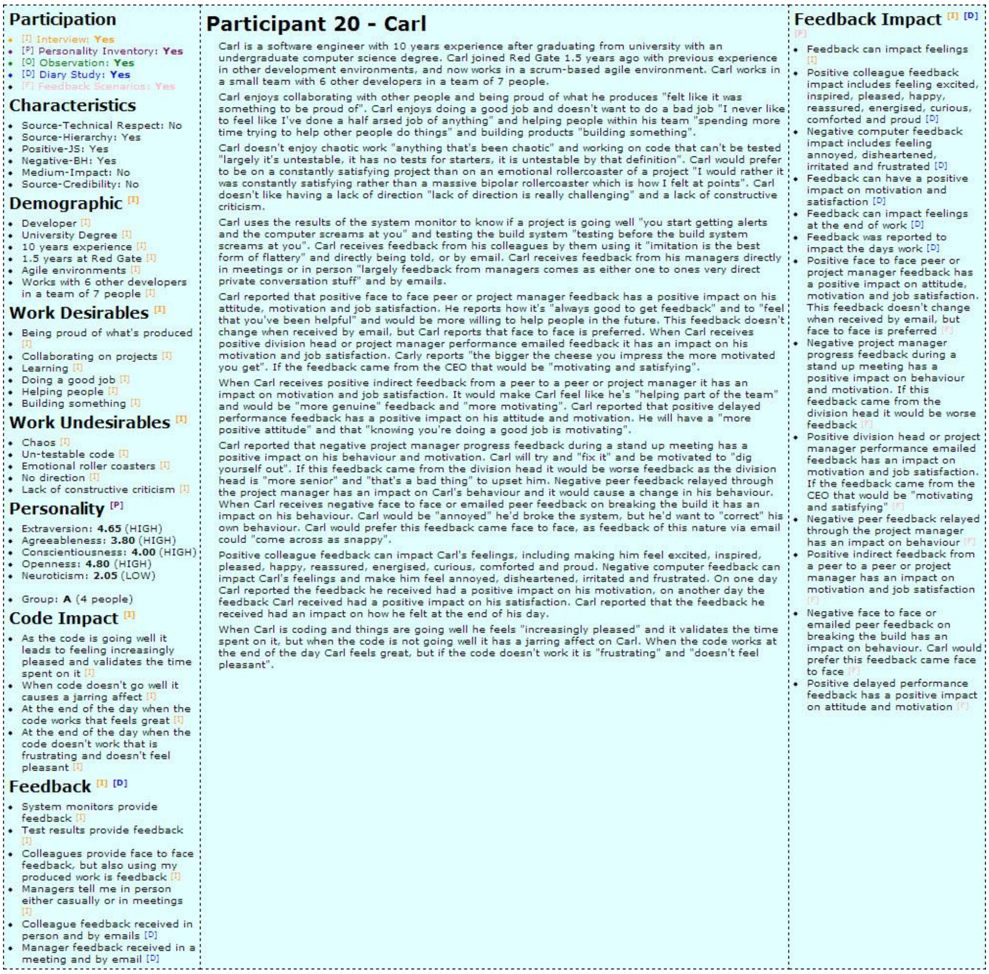
Feedback Impact [D]

- Negative feedback has a strong impact, including feeling annoyed, frustrated, disheartened, unhappy and disappointed [D]
- Positive feedback has a strong impact, including feeling happy, comforted, pleased, proud, excited and inspired [D]
- Feedback can impact feelings at the end of work [D]
- Feedback was reported to impact the days work [D]
- Feedback can have a negative impact on motivation, satisfaction and enjoyment [D]
- Feedback can have a positive impact on productivity and enjoyment [D]

Code Impact [I]

- Code going right leads to professional pride [I]
- Coding going wrong can lead to avoidance, procrastination, or starting again [I]
- When working all day and the code works, this feels pretty good and leads to feeling [I]proud [I]
- When working all day and the code doesn't work it leads to feeling crushed and wanting to go home [I]

PARTICIPANT 19 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **Yes**

Characteristics

- Source-Technical Respect: No
- Source-Hierarchy: Yes
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: No
- Source-Credibility: No

Demographic [I]

- Developer [I]
- University Degree [I]
- 10 years experience [I]
- 1.5 years at Red Gate [I]
- Agile environments [I]
- Works with 6 other developers in a team of 7 people [I]

Work Desirables [I]

- Being proud of what's produced [I]
- Collaborating on projects [I]
- Learning [I]
- Doing a good job [I]
- Helping people [I]
- Building something [I]

Work Undesirables [I]

- Chaos [I]
- Un-testable code [I]
- Emotional roller coasters [I]
- No direction [I]
- Lack of constructive criticism [I]

Personality [P]

- Extraversion: **4.65** (HIGH)
- Agreeableness: **3.80** (HIGH)
- Conscientiousness: **4.00** (HIGH)
- Openness: **4.80** (HIGH)
- Neuroticism: **2.05** (LOW)

- Group: **A** (4 people)

Code Impact [I]

- As the code is going well it leads to feeling increasingly pleased and validates the time spent on it [I]
- When code doesn't go well it causes a jarring affect [I]
- At the end of the day when the code works that feels great [I]
- At the end of the day when the code doesn't work that is frustrating and doesn't feel pleasant [I]

Feedback [I] [D]

- System monitors provide feedback [I]
- Test results provide feedback [I]
- Colleagues provide face to face feedback, but also using my produced work is feedback [I]
- Managers tell me in person either casually or in meetings [I]
- Colleague feedback received in person and by emails [D]
- Manager feedback received in a meeting and by email [D]

Participant 20 - Carl

Carl is a software engineer with 10 years experience after graduating from university with an undergraduate computer science degree. Carl joined Red Gate 1.5 years ago with previous experience in other development environments, and now works in a scrum-based agile environment. Carl works in a small team with 6 other developers in a team of 7 people.

Carl enjoys collaborating with other people and being proud of what he produces "felt like it was something to be proud of". Carl enjoys doing a good job and doesn't want to do a bad job "I never like to feel like I've done a half arsed job of anything" and helping people within his team "spending more time trying to help other people do things" and building products "building something".

Carl doesn't enjoy chaotic work "anything that's been chaotic" and working on code that can't be tested "largely it's untestable, it has no tests for starters, it is untestable by that definition". Carl would prefer to be on a constantly satisfying project than on an emotional rollercoaster of a project "I would rather it was constantly satisfying rather than a massive bipolar rollercoaster which is how I felt at points". Carl doesn't like having a lack of direction "lack of direction is really challenging" and a lack of constructive criticism.

Carl uses the results of the system monitor to know if a project is going well "you start getting alerts and the computer screams at you" and testing the build system "testing before the build system screams at you". Carl receives feedback from his colleagues by them using it "imitation is the best form of flattery" and directly being told, or by email. Carl receives feedback from his managers directly in meetings or in person "largely feedback from managers comes as either one to ones very direct private conversation stuff" and by emails.

Carl reported that positive face to face peer or project manager feedback has a positive impact on his attitude, motivation and job satisfaction. He reports how it's "always good to get feedback" and to "feel that you've been helpful" and would be more willing to help people in the future. This feedback doesn't change when received by email, but Carl reports that face to face is preferred. When Carl receives positive division head or project manager performance emailed feedback it has an impact on his motivation and job satisfaction. Carly reports "the bigger the cheese you impress the more motivated you get". If the feedback came from the CEO that would be "motivating and satisfying".

When Carl receives positive indirect feedback from a peer to a peer or project manager it has an impact on motivation and job satisfaction. It would make Carl feel like he's "helping part of the team" and would be "more genuine" feedback and "more motivating". Carl reported that positive delayed performance feedback has a positive impact on his attitude and motivation. He will have a "more positive attitude" and that "knowing you're doing a good job is motivating".

Carl reported that negative project manager progress feedback during a stand up meeting has a positive impact on his behaviour and motivation. Carl will try and "fix it" and be motivated to "dig yourself out". If this feedback came from the division head it would be worse feedback as the division head is "more senior" and "that's a bad thing" to upset him. Negative peer feedback relayed through the project manager has an impact on Carl's behaviour and it would cause a change in his behaviour. When Carl receives negative face to face or emailed peer feedback on breaking the build it has an impact on his behaviour. Carl would be "annoyed" he'd broke the system, but he'd want to "correct" his own behaviour. Carl would prefer this feedback came face to face, as feedback of this nature via email could "come across as snappy".

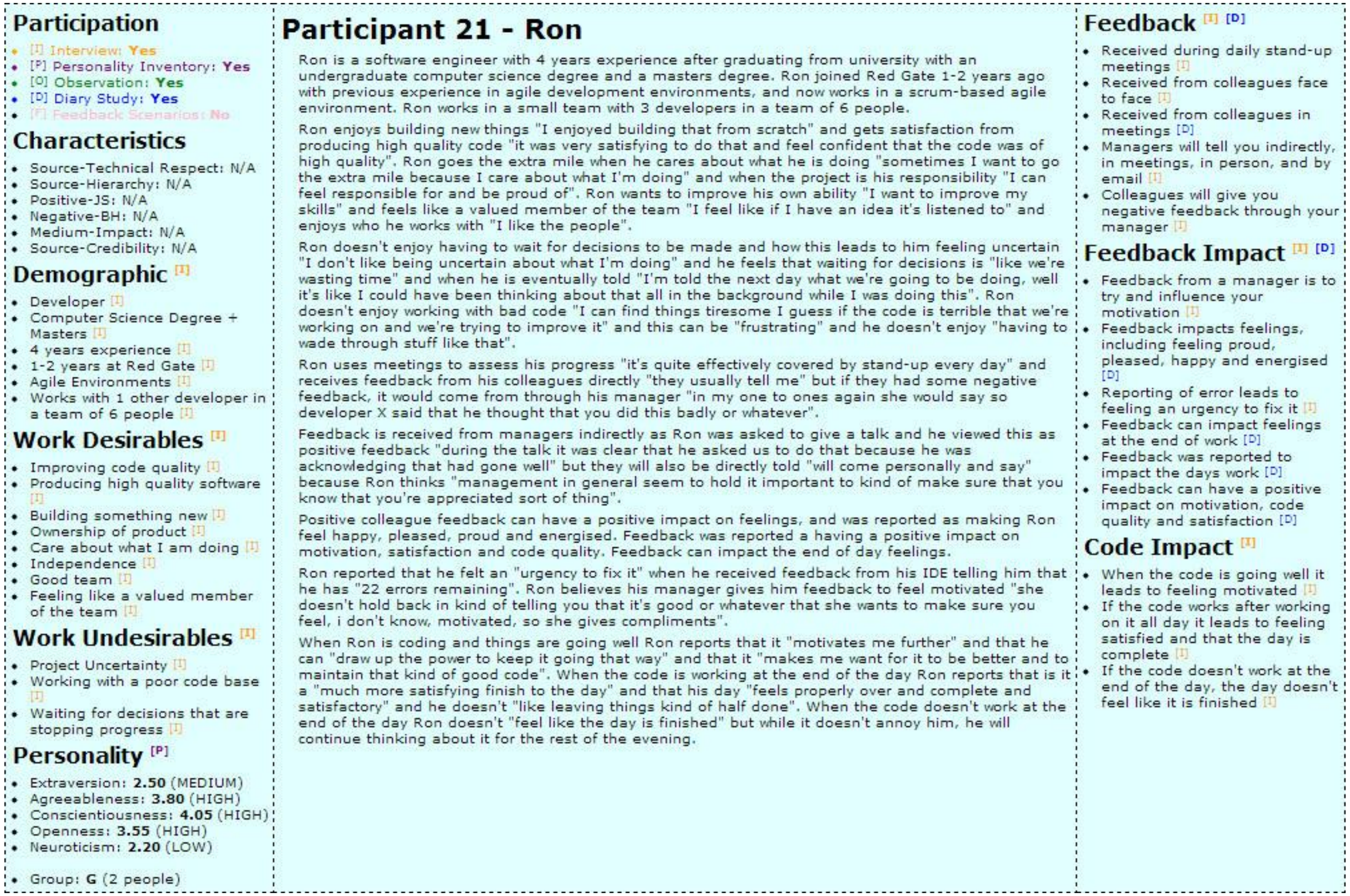
Positive colleague feedback can impact Carl's feelings, including making him feel excited, inspired, pleased, happy, reassured, energised, curious, comforted and proud. Negative computer feedback can impact Carl's feelings and make him feel annoyed, disheartened, irritated and frustrated. On one day Carl reported the feedback he received had a positive impact on his motivation, on another day the feedback Carl received had a positive impact on his satisfaction. Carl reported that the feedback he received had an impact on how he felt at the end of his day.

When Carl is coding and things are going well he feels "increasingly pleased" and it validates the time spent on it, but when the code is not going well it has a jarring affect on Carl. When the code works at the end of the day Carl feels great, but if the code doesn't work it is "frustrating" and "doesn't feel pleasant".

Feedback Impact [I] [D] [F]

- Feedback can impact feelings [I]
- Positive colleague feedback impact includes feeling excited, inspired, pleased, happy, reassured, energised, curious, comforted and proud [D]
- Negative computer feedback impact includes feeling annoyed, disheartened, irritated and frustrated [D]
- Feedback can have a positive impact on motivation and satisfaction [D]
- Feedback can impact feelings at the end of work [D]
- Feedback was reported to impact the days work [D]
- Positive face to face peer or project manager feedback has a positive impact on attitude, motivation and job satisfaction. This feedback doesn't change when received by email, but face to face is preferred [F]
- Negative project manager progress feedback during a stand up meeting has a positive impact on behaviour and motivation. If this feedback came from the division head it would be worse feedback [F]
- Positive division head or project manager performance emailed feedback has an impact on motivation and job satisfaction. If the feedback came from the CEO that would be "motivating and satisfying" [F]
- Negative peer feedback relayed through the project manager has an impact on behaviour [F]
- Positive indirect feedback from a peer to a peer or project manager has an impact on motivation and job satisfaction [F]
- Negative face to face or emailed peer feedback on breaking the build has an impact on behaviour. Carl would prefer this feedback came face to face [F]
- Positive delayed performance feedback has a positive impact on attitude and motivation [F]

PARTICIPANT 20 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **No**

Characteristics

- Source-Technical Respect: N/A
- Source-Hierarchy: N/A
- Positive-JS: N/A
- Negative-BH: N/A
- Medium-Impact: N/A
- Source-Credibility: N/A

Demographic [I]

- Developer [I]
- Computer Science Degree + Masters [I]
- 4 years experience [I]
- 1-2 years at Red Gate [I]
- Agile Environments [I]
- Works with 1 other developer in a team of 6 people [I]

Work Desirables [I]

- Improving code quality [I]
- Producing high quality software [I]
- Building something new [I]
- Ownership of product [I]
- Care about what I am doing [I]
- Independence [I]
- Good team [I]
- Feeling like a valued member of the team [I]

Work Undesirables [I]

- Project Uncertainty [I]
- Working with a poor code base [I]
- Waiting for decisions that are stopping progress [I]

Personality [P]

- Extraversion: **2.50** (MEDIUM)
- Agreeableness: **3.80** (HIGH)
- Conscientiousness: **4.05** (HIGH)
- Openness: **3.55** (HIGH)
- Neuroticism: **2.20** (LOW)

- Group: **G** (2 people)

Participant 21 - Ron

Ron is a software engineer with 4 years experience after graduating from university with an undergraduate computer science degree and a masters degree. Ron joined Red Gate 1-2 years ago with previous experience in agile development environments, and now works in a scrum-based agile environment. Ron works in a small team with 3 developers in a team of 6 people.

Ron enjoys building new things "I enjoyed building that from scratch" and gets satisfaction from producing high quality code "it was very satisfying to do that and feel confident that the code was of high quality". Ron goes the extra mile when he cares about what he is doing "sometimes I want to go the extra mile because I care about what I'm doing" and when the project is his responsibility "I can feel responsible for and be proud of". Ron wants to improve his own ability "I want to improve my skills" and feels like a valued member of the team "I feel like if I have an idea it's listened to" and enjoys who he works with "I like the people".

Ron doesn't enjoy having to wait for decisions to be made and how this leads to him feeling uncertain "I don't like being uncertain about what I'm doing" and he feels that waiting for decisions is "like we're wasting time" and when he is eventually told "I'm told the next day what we're going to be doing, well it's like I could have been thinking about that all in the background while I was doing this". Ron doesn't enjoy working with bad code "I can find things tiresome I guess if the code is terrible that we're working on and we're trying to improve it" and this can be "frustrating" and he doesn't enjoy "having to wade through stuff like that".

Ron uses meetings to assess his progress "it's quite effectively covered by stand-up every day" and receives feedback from his colleagues directly "they usually tell me" but if they had some negative feedback, it would come from through his manager "in my one to ones again she would say so developer X said that he thought that you did this badly or whatever".

Feedback is received from managers indirectly as Ron was asked to give a talk and he viewed this as positive feedback "during the talk it was clear that he asked us to do that because he was acknowledging that had gone well" but they will also be directly told "will come personally and say" because Ron thinks "management in general seem to hold it important to kind of make sure that you know that you're appreciated sort of thing".

Positive colleague feedback can have a positive impact on feelings, and was reported as making Ron feel happy, pleased, proud and energised. Feedback was reported a having a positive impact on motivation, satisfaction and code quality. Feedback can impact the end of day feelings.

Ron reported that he felt an "urgency to fix it" when he received feedback from his IDE telling him that he has "22 errors remaining". Ron believes his manager gives him feedback to feel motivated "she doesn't hold back in kind of telling you that it's good or whatever that she wants to make sure you feel, i don't know, motivated, so she gives compliments".

When Ron is coding and things are going well Ron reports that it "motivates me further" and that he can "draw up the power to keep it going that way" and that it "makes me want for it to be better and to maintain that kind of good code". When the code is working at the end of the day Ron reports that is it a "much more satisfying finish to the day" and that his day "feels properly over and complete and satisfactory" and he doesn't "like leaving things kind of half done". When the code doesn't work at the end of the day Ron doesn't "feel like the day is finished" but while it doesn't annoy him, he will continue thinking about it for the rest of the evening.

Feedback [I] [D]

- Received during daily stand-up meetings [I]
- Received from colleagues face to face [I]
- Received from colleagues in meetings [D]
- Managers will tell you indirectly, in meetings, in person, and by email [I]
- Colleagues will give you negative feedback through your manager [I]

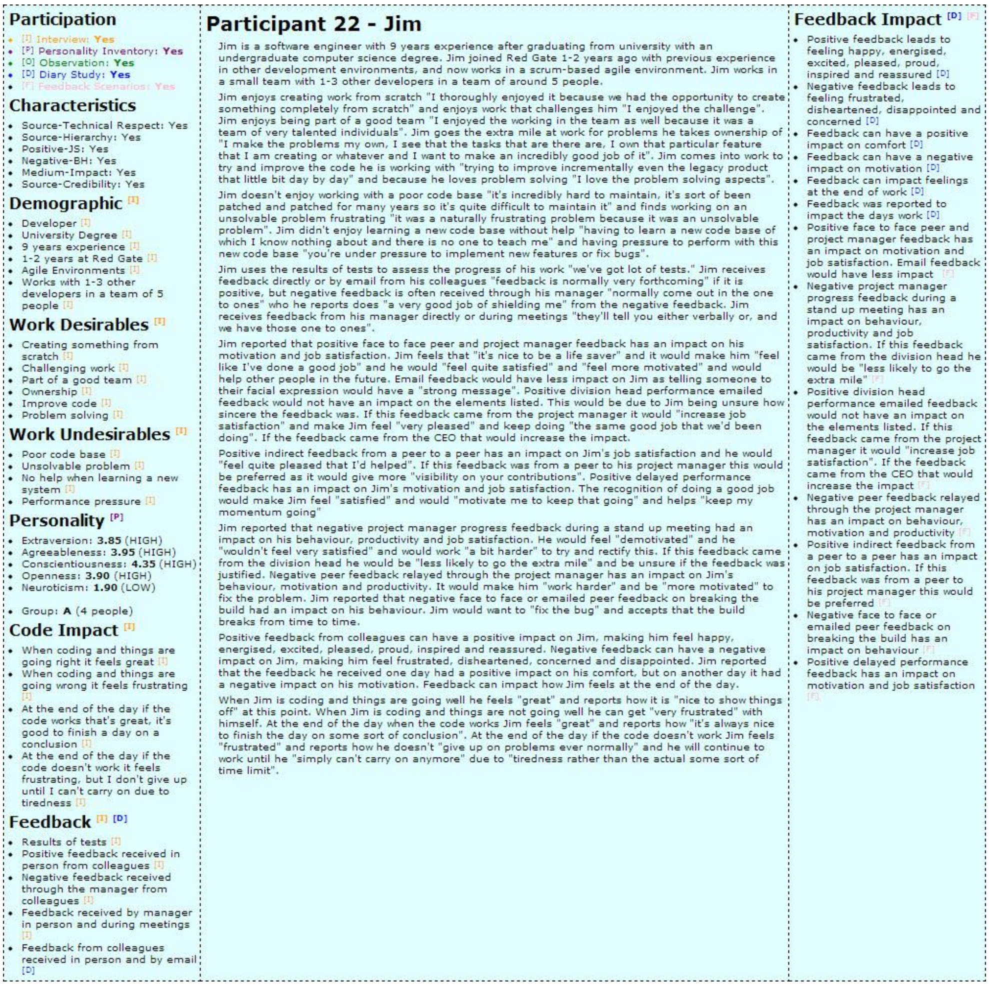
Feedback Impact [I] [D]

- Feedback from a manager is to try and influence your motivation [I]
- Feedback impacts feelings, including feeling proud, pleased, happy and energised [D]
- Reporting of error leads to feeling an urgency to fix it [I]
- Feedback can impact feelings at the end of work [D]
- Feedback was reported to impact the days work [D]
- Feedback can have a positive impact on motivation, code quality and satisfaction [D]

Code Impact [I]

- When the code is going well it leads to feeling motivated [I]
- If the code works after working on it all day it leads to feeling satisfied and that the day is complete [I]
- If the code doesn't work at the end of the day, the day doesn't feel like it is finished [I]

PARTICIPANT 21 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **Yes**

Characteristics

- Source-Technical Respect: Yes
- Source-Hierarchy: Yes
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: Yes
- Source-Credibility: Yes

Demographic [I]

- Developer [I]
- University Degree [I]
- 9 years experience [I]
- 1-2 years at Red Gate [I]
- Agile Environments [I]
- Works with 1-3 other developers in a team of 5 people [I]

Work Desirables [I]

- Creating something from scratch [I]
- Challenging work [I]
- Part of a good team [I]
- Ownership [I]
- Improve code [I]
- Problem solving [I]

Work Undesirables [I]

- Poor code base [I]
- Unsolvable problem [I]
- No help when learning a new system [I]
- Performance pressure [I]

Personality [P]

- Extraversion: **3.85** (HIGH)
- Agreeableness: **3.95** (HIGH)
- Conscientiousness: **4.35** (HIGH)
- Openness: **3.90** (HIGH)
- Neuroticism: **1.90** (LOW)

- Group: **A** (4 people)

Code Impact [I]

- When coding and things are going right it feels great [I]
- When coding and things are going wrong it feels frustrating [I]
- At the end of the day if the code works that's great, it's good to finish a day on a conclusion [I]
- At the end of the day if the code doesn't work it feels frustrating, but I don't give up until I can't carry on due to tiredness [I]

Feedback [I] [D]

- Results of tests [I]
- Positive feedback received in person from colleagues [I]
- Negative feedback received through the manager from colleagues [I]
- Feedback received by manager in person and during meetings [I]
- Feedback from colleagues received in person and by email [D]

Participant 22 - Jim

Jim is a software engineer with 9 years experience after graduating from university with an undergraduate computer science degree. Jim joined Red Gate 1-2 years ago with previous experience in other development environments, and now works in a scrum-based agile environment. Jim works in a small team with 1-3 other developers in a team of around 5 people.

Jim enjoys creating work from scratch "I thoroughly enjoyed it because we had the opportunity to create something completely from scratch" and enjoys work that challenges him "I enjoyed the challenge". Jim enjoys being part of a good team "I enjoyed the working in the team as well because it was a team of very talented individuals". Jim goes the extra mile at work for problems he takes ownership of "I make the problems my own, I see that the tasks that are there are, I own that particular feature that I am creating or whatever and I want to make an incredibly good job of it". Jim comes into work to try and improve the code he is working with "trying to improve incrementally even the legacy product that little bit day by day" and because he loves problem solving "I love the problem solving aspects".

Jim doesn't enjoy working with a poor code base "it's incredibly hard to maintain, it's sort of been patched and patched for many years so it's quite difficult to maintain it" and finds working on an unsolvable problem frustrating "it was a naturally frustrating problem because it was an unsolvable problem". Jim didn't enjoy learning a new code base without help "having to learn a new code base of which I know nothing about and there is no one to teach me" and having pressure to perform with this new code base "you're under pressure to implement new features or fix bugs".

Jim uses the results of tests to assess the progress of his work "we've got lot of tests." Jim receives feedback directly or by email from his colleagues "feedback is normally very forthcoming" if it is positive, but negative feedback is often received through his manager "normally come out in the one to ones" who he reports does "a very good job of shielding me" from the negative feedback. Jim receives feedback from his manager directly or during meetings "they'll tell you either verbally or, and we have those one to ones".

Jim reported that positive face to face peer and project manager feedback has an impact on his motivation and job satisfaction. Jim feels that "it's nice to be a life saver" and it would make him "feel like I've done a good job" and he would "feel quite satisfied" and "feel more motivated" and would help other people in the future. Email feedback would have less impact on Jim as telling someone to their facial expression would have a "strong message". Positive division head performance emailed feedback would not have an impact on the elements listed. This would be due to Jim being unsure how sincere the feedback was. If this feedback came from the project manager it would "increase job satisfaction" and make Jim feel "very pleased" and keep doing "the same good job that we'd been doing". If the feedback came from the CEO that would increase the impact.

Positive indirect feedback from a peer to a peer has an impact on Jim's job satisfaction and he would "feel quite pleased that I'd helped". If this feedback was from a peer to his project manager this would be preferred as it would give more "visibility on your contributions". Positive delayed performance feedback has an impact on Jim's motivation and job satisfaction. The recognition of doing a good job would make Jim feel "satisfied" and would "motivate me to keep that going" and helps "keep my momentum going"

Jim reported that negative project manager progress feedback during a stand up meeting had an impact on his behaviour, productivity and job satisfaction. He would feel "demotivated" and he "wouldn't feel very satisfied" and would work "a bit harder" to try and rectify this. If this feedback came from the division head he would be "less likely to go the extra mile" and be unsure if the feedback was justified. Negative peer feedback relayed through the project manager has an impact on Jim's behaviour, motivation and productivity. It would make him "work harder" and be "more motivated" to fix the problem. Jim reported that negative face to face or emailed peer feedback on breaking the build had an impact on his behaviour. Jim would want to "fix the bug" and accepts that the build breaks from time to time.

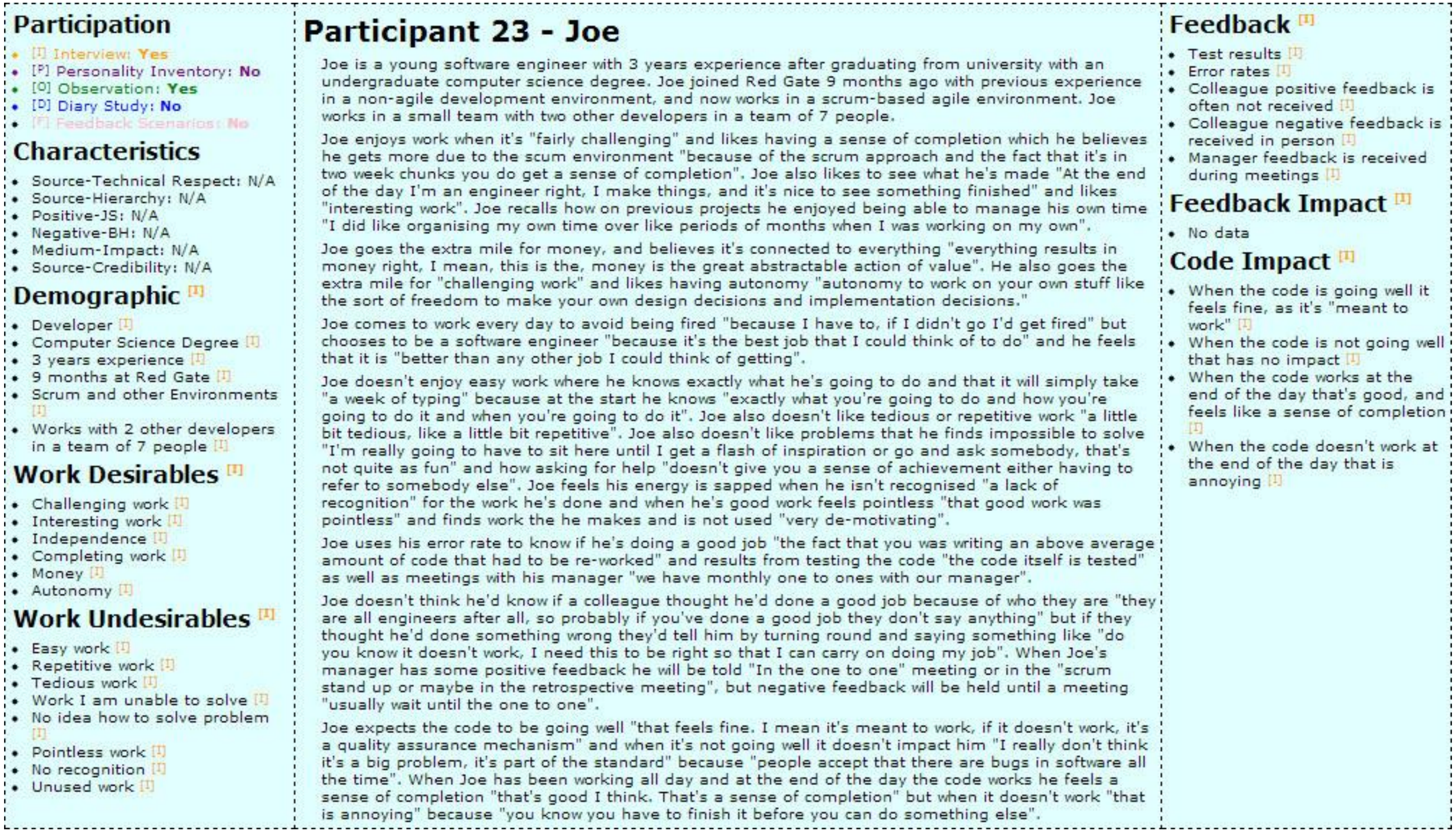
Positive feedback from colleagues can have a positive impact on Jim, making him feel happy, energised, excited, pleased, proud, inspired and reassured. Negative feedback can have a negative impact on Jim, making him feel frustrated, disheartened, concerned and disappointed. Jim reported that the feedback he received one day had a positive impact on his comfort, but on another day it had a negative impact on his motivation. Feedback can impact how Jim feels at the end of the day.

When Jim is coding and things are going well he feels "great" and reports how it is "nice to show things off" at this point. When Jim is coding and things are not going well he can get "very frustrated" with himself. At the end of the day when the code works Jim feels "great" and reports how "it's always nice to finish the day on some sort of conclusion". At the end of the day if the code doesn't work Jim feels "frustrated" and reports how he doesn't "give up on problems ever normally" and he will continue to work until he "simply can't carry on anymore" due to "tiredness rather than the actual some sort of time limit".

Feedback Impact [D] [F]

- Positive feedback leads to feeling happy, energised, excited, pleased, proud, inspired and reassured [D]
- Negative feedback leads to feeling frustrated, disheartened, disappointed and concerned [D]
- Feedback can have a positive impact on comfort [D]
- Feedback can have a negative impact on motivation [D]
- Feedback can impact feelings at the end of work [D]
- Feedback was reported to impact the days work [D]
- Positive face to face peer and project manager feedback has an impact on motivation and job satisfaction. Email feedback would have less impact [F]
- Negative project manager progress feedback during a stand up meeting has an impact on behaviour, productivity and job satisfaction. If this feedback came from the division head he would be "less likely to go the extra mile" [F]
- Positive division head performance emailed feedback would not have an impact on the elements listed. If this feedback came from the project manager it would "increase job satisfaction". If the feedback came from the CEO that would increase the impact [F]
- Negative peer feedback relayed through the project manager has an impact on behaviour, motivation and productivity [F]
- Positive indirect feedback from a peer to a peer has an impact on job satisfaction. If this feedback was from a peer to his project manager this would be preferred [F]
- Negative face to face or emailed peer feedback on breaking the build has an impact on behaviour [F]
- Positive delayed performance feedback has an impact on motivation and job satisfaction [F]

PARTICIPANT 22 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **No**
- [O] Observation: **Yes**
- [D] Diary Study: **No**
- [F] Feedback Scenarios: **No**

Characteristics

- Source-Technical Respect: N/A
- Source-Hierarchy: N/A
- Positive-JS: N/A
- Negative-BH: N/A
- Medium-Impact: N/A
- Source-Credibility: N/A

Demographic [I]

- Developer [I]
- Computer Science Degree [I]
- 3 years experience [I]
- 9 months at Red Gate [I]
- Scrum and other Environments [I]
- Works with 2 other developers in a team of 7 people [I]

Work Desirables [I]

- Challenging work [I]
- Interesting work [I]
- Independence [I]
- Completing work [I]
- Money [I]
- Autonomy [I]

Work Undesirables [I]

- Easy work [I]
- Repetitive work [I]
- Tedious work [I]
- Work I am unable to solve [I]
- No idea how to solve problem [I]
- Pointless work [I]
- No recognition [I]
- Unused work [I]

Participant 23 - Joe

Joe is a young software engineer with 3 years experience after graduating from university with an undergraduate computer science degree. Joe joined Red Gate 9 months ago with previous experience in a non-agile development environment, and now works in a scrum-based agile environment. Joe works in a small team with two other developers in a team of 7 people.

Joe enjoys work when it's "fairly challenging" and likes having a sense of completion which he believes he gets more due to the scum environment "because of the scrum approach and the fact that it's in two week chunks you do get a sense of completion". Joe also likes to see what he's made "At the end of the day I'm an engineer right, I make things, and it's nice to see something finished" and likes "interesting work". Joe recalls how on previous projects he enjoyed being able to manage his own time "I did like organising my own time over like periods of months when I was working on my own".

Joe goes the extra mile for money, and believes it's connected to everything "everything results in money right, I mean, this is the, money is the great abstractable action of value". He also goes the extra mile for "challenging work" and likes having autonomy "autonomy to work on your own stuff like the sort of freedom to make your own design decisions and implementation decisions."

Joe comes to work every day to avoid being fired "because I have to, if I didn't go I'd get fired" but chooses to be a software engineer "because it's the best job that I could think of to do" and he feels that it is "better than any other job I could think of getting".

Joe doesn't enjoy easy work where he knows exactly what he's going to do and that it will simply take "a week of typing" because at the start he knows "exactly what you're going to do and how you're going to do it and when you're going to do it". Joe also doesn't like tedious or repetitive work "a little bit tedious, like a little bit repetitive". Joe also doesn't like problems that he finds impossible to solve "I'm really going to have to sit here until I get a flash of inspiration or go and ask somebody, that's not quite as fun" and how asking for help "doesn't give you a sense of achievement either having to refer to somebody else". Joe feels his energy is sapped when he isn't recognised "a lack of recognition" for the work he's done and when he's good work feels pointless "that good work was pointless" and finds work the he makes and is not used "very de-motivating".

Joe uses his error rate to know if he's doing a good job "the fact that you was writing an above average amount of code that had to be re-worked" and results from testing the code "the code itself is tested" as well as meetings with his manager "we have monthly one to ones with our manager".

Joe doesn't think he'd know if a colleague thought he'd done a good job because of who they are "they are all engineers after all, so probably if you've done a good job they don't say anything" but if they thought he'd done something wrong they'd tell him by turning round and saying something like "do you know it doesn't work, I need this to be right so that I can carry on doing my job". When Joe's manager has some positive feedback he will be told "In the one to one" meeting or in the "scrum stand up or maybe in the retrospective meeting", but negative feedback will be held until a meeting "usually wait until the one to one".

Joe expects the code to be going well "that feels fine. I mean it's meant to work, if it doesn't work, it's a quality assurance mechanism" and when it's not going well it doesn't impact him "I really don't think it's a big problem, it's part of the standard" because "people accept that there are bugs in software all the time". When Joe has been working all day and at the end of the day the code works he feels a sense of completion "that's good I think. That's a sense of completion" but when it doesn't work "that is annoying" because "you know you have to finish it before you can do something else".

Feedback [I]

- Test results [I]
- Error rates [I]
- Colleague positive feedback is often not received [I]
- Colleague negative feedback is received in person [I]
- Manager feedback is received during meetings [I]

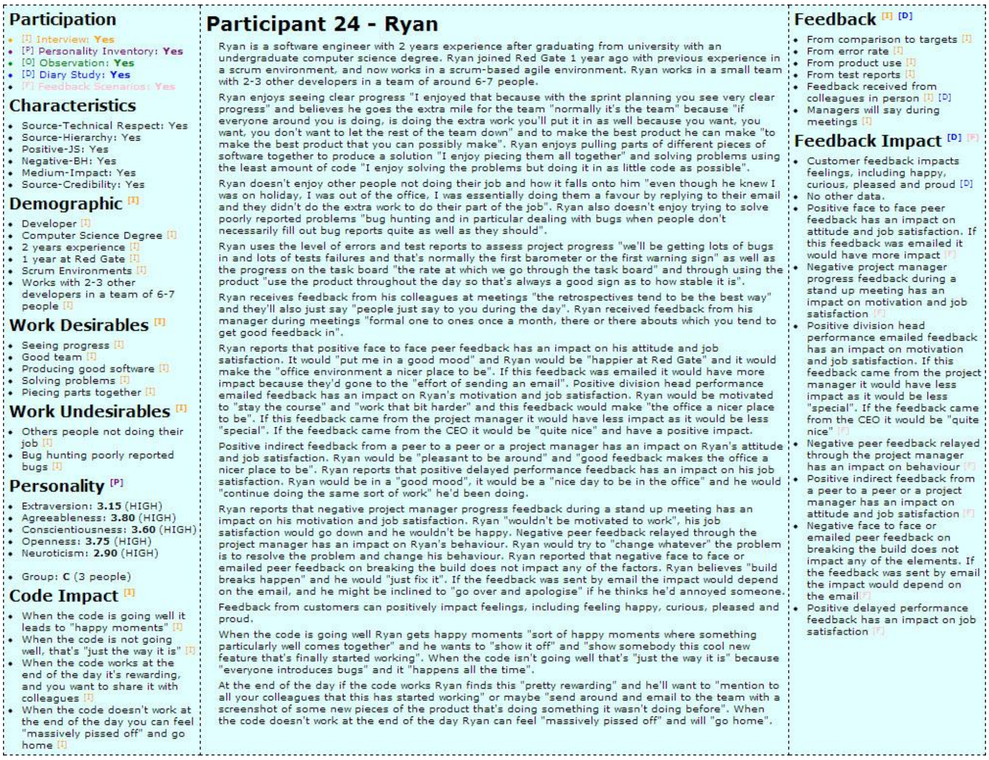
Feedback Impact [I]

- No data

Code Impact [I]

- When the code is going well it feels fine, as it's "meant to work" [I]
- When the code is not going well that has no impact [I]
- When the code works at the end of the day that's good, and feels like a sense of completion [I]
- When the code doesn't work at the end of the day that is annoying [I]

PARTICIPANT 23 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **Yes**

Characteristics

- Source-Technical Respect: Yes
- Source-Hierarchy: Yes
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: Yes
- Source-Credibility: Yes

Demographic [I]

- Developer [I]
- Computer Science Degree [I]
- 2 years experience [I]
- 1 year at Red Gate [I]
- Scrum Environments [I]
- Works with 2-3 other developers in a team of 6-7 people [I]

Work Desirables [I]

- Seeing progress [I]
- Good team [I]
- Producing good software [I]
- Solving problems [I]
- Piecing parts together [I]

Work Undesirables [I]

- Others people not doing their job [I]
- Bug hunting poorly reported bugs [I]

Personality [P]

- Extraversion: **3.15** (HIGH)
- Agreeableness: **3.80** (HIGH)
- Conscientiousness: **3.60** (HIGH)
- Openness: **3.75** (HIGH)
- Neuroticism: **2.90** (HIGH)

- Group: **C** (3 people)

Code Impact [I]

- When the code is going well it leads to "happy moments" [I]
- When the code is not going well, that's "just the way it is" [I]
- When the code works at the end of the day it's rewarding, and you want to share it with colleagues [I]
- When the code doesn't work at the end of the day you can feel "massively pissed off" and go home [I]

Participant 24 - Ryan

Ryan is a software engineer with 2 years experience after graduating from university with an undergraduate computer science degree. Ryan joined Red Gate 1 year ago with previous experience in a scrum environment, and now works in a scrum-based agile environment. Ryan works in a small team with 2-3 other developers in a team of around 6-7 people.

Ryan enjoys seeing clear progress "I enjoyed that because with the sprint planning you see very clear progress" and believes he goes the extra mile for the team "normally it's the team" because "if everyone around you is doing, is doing the extra work you'll put it in as well because you want, you want, you don't want to let the rest of the team down" and to make the best product he can make "to make the best product that you can possibly make". Ryan enjoys pulling parts of different pieces of software together to produce a solution "I enjoy piecing them all together" and solving problems using the least amount of code "I enjoy solving the problems but doing it in as little code as possible".

Ryan doesn't enjoy other people not doing their job and how it falls onto him "even though he knew I was on holiday, I was out of the office, I was essentially doing them a favour by replying to their email and they didn't do the extra work to do their part of the job". Ryan also doesn't enjoy trying to solve poorly reported problems "bug hunting and in particular dealing with bugs when people don't necessarily fill out bug reports quite as well as they should".

Ryan uses the level of errors and test reports to assess project progress "we'll be getting lots of bugs in and lots of tests failures and that's normally the first barometer or the first warning sign" as well as the progress on the task board "the rate at which we go through the task board" and through using the product "use the product throughout the day so that's always a good sign as to how stable it is".

Ryan receives feedback from his colleagues at meetings "the retrospectives tend to be the best way" and they'll also just say "people just say to you during the day". Ryan received feedback from his manager during meetings "formal one to ones once a month, there or there abouts which you tend to get good feedback in".

Ryan reports that positive face to face peer feedback has an impact on his attitude and job satisfaction. It would "put me in a good mood" and Ryan would be "happier at Red Gate" and it would make the "office environment a nicer place to be". If this feedback was emailed it would have more impact because they'd gone to the "effort of sending an email". Positive division head performance emailed feedback has an impact on Ryan's motivation and job satisfaction. Ryan would be motivated to "stay the course" and "work that bit harder" and this feedback would make "the office a nicer place to be". If this feedback came from the project manager it would have less impact as it would be less "special". If the feedback came from the CEO it would be "quite nice" and have a positive impact.

Positive indirect feedback from a peer to a peer or a project manager has an impact on Ryan's attitude and job satisfaction. Ryan would be "pleasant to be around" and "good feedback makes the office a nicer place to be". Ryan reports that positive delayed performance feedback has an impact on his job satisfaction. Ryan would be in a "good mood", it would be a "nice day to be in the office" and he would "continue doing the same sort of work" he'd been doing.

Ryan reports that negative project manager progress feedback during a stand up meeting has an impact on his motivation and job satisfaction. Ryan "wouldn't be motivated to work", his job satisfaction would go down and he wouldn't be happy. Negative peer feedback relayed through the project manager has an impact on Ryan's behaviour. Ryan would try to "change whatever" the problem is to resolve the problem and change his behaviour. Ryan reported that negative face to face or emailed peer feedback on breaking the build does not impact any of the factors. Ryan believes "build breaks happen" and he would "just fix it". If the feedback was sent by email the impact would depend on the email, and he might be inclined to "go over and apologise" if he thinks he'd annoyed someone.

Feedback from customers can positively impact feelings, including feeling happy, curious, pleased and proud.

When the code is going well Ryan gets happy moments "sort of happy moments where something particularly well comes together" and he wants to "show it off" and "show somebody this cool new feature that's finally started working". When the code isn't going well that's "just the way it is" because "everyone introduces bugs" and it "happens all the time".

At the end of the day if the code works Ryan finds this "pretty rewarding" and he'll want to "mention to all your colleagues that this has started working" or maybe "send around and email to the team with a screenshot of some new pieces of the product that's doing something it wasn't doing before". When the code doesn't work at the end of the day Ryan can feel "massively pissed off" and will "go home".

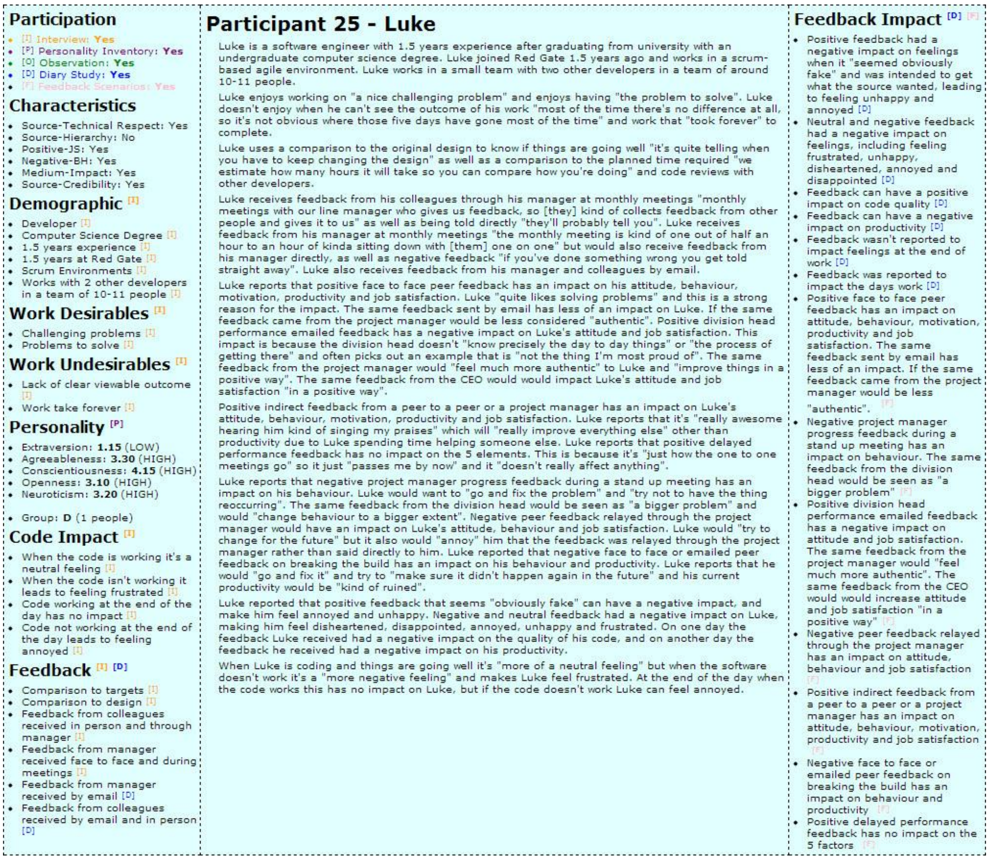
Feedback [I] [D]

- From comparison to targets [I]
- From error rate [I]
- From product use [I]
- From test reports [I]
- Feedback received from colleagues in person [I] [D]
- Managers will say during meetings [I]

Feedback Impact [D] [F]

- Customer feedback impacts feelings, including happy, curious, pleased and proud [D]
- No other data.
- Positive face to face peer feedback has an impact on attitude and job satisfaction. If this feedback was emailed it would have more impact [F]
- Negative project manager progress feedback during a stand up meeting has an impact on motivation and job satisfaction [F]
- Positive division head performance emailed feedback has an impact on motivation and job satisfaction. If this feedback came from the project manager it would have less impact as it would be less "special". If the feedback came from the CEO it would be "quite nice" [F]
- Negative peer feedback relayed through the project manager has an impact on behaviour [F]
- Positive indirect feedback from a peer to a peer or a project manager has an impact on attitude and job satisfaction [F]
- Negative face to face or emailed peer feedback on breaking the build does not impact any of the elements. If the feedback was sent by email the impact would depend on the email [F]
- Positive delayed performance feedback has an impact on job satisfaction [F]

PARTICIPANT 24 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **Yes**

Characteristics

- Source-Technical Respect: Yes
- Source-Hierarchy: No
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: Yes
- Source-Credibility: Yes

Demographic [I]

- Developer [I]
- Computer Science Degree [I]
- 1.5 years experience [I]
- 1.5 years at Red Gate [I]
- Scrum Environments [I]
- Works with 2 other developers in a team of 10-11 people [I]

Work Desirables [I]

- Challenging problems [I]
- Problems to solve [I]

Work Undesirables [I]

- Lack of clear viewable outcome [I]
- Work take forever [I]

Personality [P]

- Extraversion: **1.15** (LOW)
- Agreeableness: **3.30** (HIGH)
- Conscientiousness: **4.15** (HIGH)
- Openness: **3.10** (HIGH)
- Neuroticism: **3.20** (HIGH)

- Group: **D** (1 people)

Code Impact [I]

- When the code is working it's a neutral feeling [I]
- When the code isn't working it leads to feeling frustrated [I]
- Code working at the end of the day has no impact [I]
- Code not working at the end of the day leads to feeling annoyed [I]

Feedback [I] [D]

- Comparison to targets [I]
- Comparison to design [I]
- Feedback from colleagues received in person and through manager [I]
- Feedback from manager received face to face and during meetings [I]
- Feedback from manager received by email [D]
- Feedback from colleagues received by email and in person [D]

Participant 25 - Luke

Luke is a software engineer with 1.5 years experience after graduating from university with an undergraduate computer science degree. Luke joined Red Gate 1.5 years ago and works in a scrum-based agile environment. Luke works in a small team with two other developers in a team of around 10-11 people.

Luke enjoys working on "a nice challenging problem" and enjoys having "the problem to solve". Luke doesn't enjoy when he can't see the outcome of his work "most of the time there's no difference at all, so it's not obvious where those five days have gone most of the time" and work that "took forever" to complete.

Luke uses a comparison to the original design to know if things are going well "it's quite telling when you have to keep changing the design" as well as a comparison to the planned time required "we estimate how many hours it will take so you can compare how you're doing" and code reviews with other developers.

Luke receives feedback from his colleagues through his manager at monthly meetings "monthly meetings with our line manager who gives us feedback, so [they] kind of collects feedback from other people and gives it to us" as well as being told directly "they'll probably tell you". Luke receives feedback from his manager at monthly meetings "the monthly meeting is kind of one out of half an hour to an hour of kinda sitting down with [them] one on one" but would also receive feedback from his manager directly, as well as negative feedback "if you've done something wrong you get told straight away". Luke also receives feedback from his manager and colleagues by email.

Luke reports that positive face to face peer feedback has an impact on his attitude, behaviour, motivation, productivity and job satisfaction. Luke "quite likes solving problems" and this is a strong reason for the impact. The same feedback sent by email has less of an impact on Luke. If the same feedback came from the project manager would be less considered "authentic". Positive division head performance emailed feedback has a negative impact on Luke's attitude and job satisfaction. This impact is because the division head doesn't "know precisely the day to day things" or "the process of getting there" and often picks out an example that is "not the thing I'm most proud of". The same feedback from the project manager would "feel much more authentic" to Luke and "improve things in a positive way". The same feedback from the CEO would would impact Luke's attitude and job satisfaction "in a positive way".

Positive indirect feedback from a peer to a peer or a project manager has an impact on Luke's attitude, behaviour, motivation, productivity and job satisfaction. Luke reports that it's "really awesome hearing him kind of singing my praises" which will "really improve everything else" other than productivity due to Luke spending time helping someone else. Luke reports that positive delayed performance feedback has no impact on the 5 elements. This is because it's "just how the one to one meetings go" so it just "passes me by now" and it "doesn't really affect anything".

Luke reports that negative project manager progress feedback during a stand up meeting has an impact on his behaviour. Luke would want to "go and fix the problem" and "try not to have the thing reoccurring". The same feedback from the division head would be seen as "a bigger problem" and would "change behaviour to a bigger extent". Negative peer feedback relayed through the project manager would have an impact on Luke's attitude, behaviour and job satisfaction. Luke would "try to change for the future" but it also would "annoy" him that the feedback was relayed through the project manager rather than said directly to him. Luke reported that negative face to face or emailed peer feedback on breaking the build has an impact on his behaviour and productivity. Luke reports that he would "go and fix it" and try to "make sure it didn't happen again in the future" and his current productivity would be "kind of ruined".

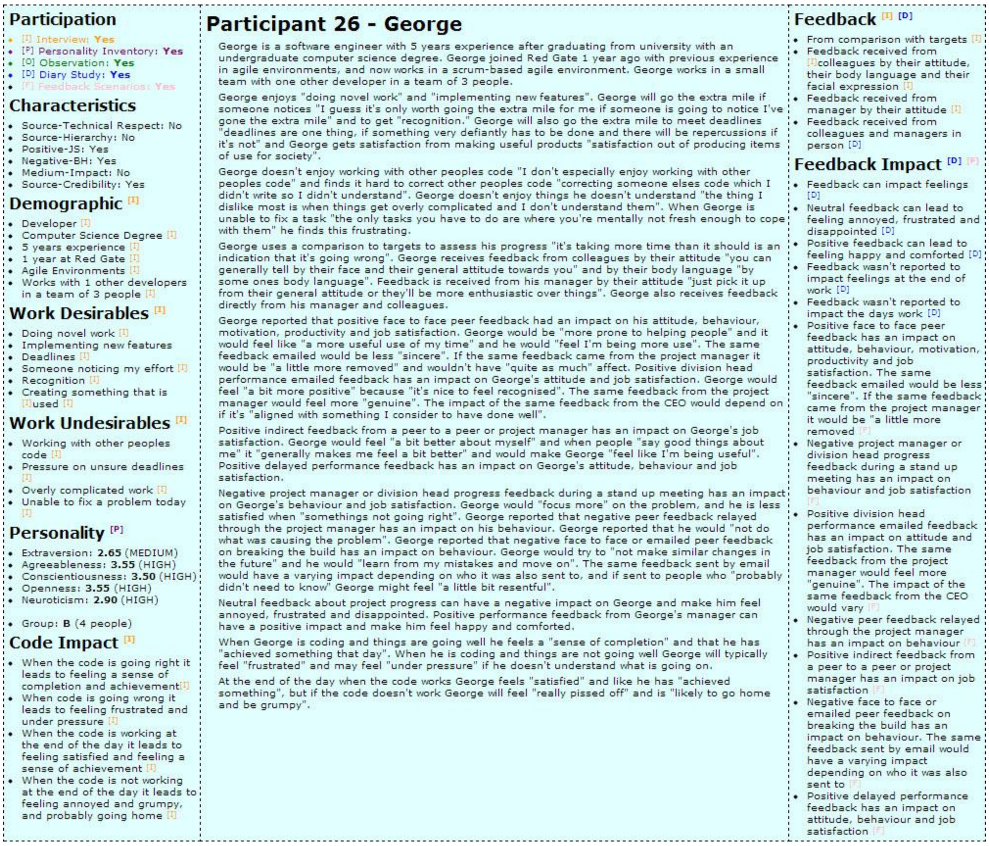
Luke reported that positive feedback that seems "obviously fake" can have a negative impact, and make him feel annoyed and unhappy. Negative and neutral feedback had a negative impact on Luke, making him feel disheartened, disappointed, annoyed, unhappy and frustrated. On one day the feedback Luke received had a negative impact on the quality of his code, and on another day the feedback he received had a negative impact on his productivity.

When Luke is coding and things are going well it's "more of a neutral feeling" but when the software doesn't work it's a "more negative feeling" and makes Luke feel frustrated. At the end of the day when the code works this has no impact on Luke, but if the code doesn't work Luke can feel annoyed.

Feedback Impact [D] [F]

- Positive feedback had a negative impact on feelings when it "seemed obviously fake" and was intended to get what the source wanted, leading to feeling unhappy and annoyed [D]
- Neutral and negative feedback had a negative impact on feelings, including feeling frustrated, unhappy, disheartened, annoyed and disappointed [D]
- Feedback can have a positive impact on code quality [D]
- Feedback can have a negative impact on productivity [D]
- Feedback wasn't reported to impact feelings at the end of work [D]
- Feedback was reported to impact the days work [D]
- Positive face to face peer feedback has an impact on attitude, behaviour, motivation, productivity and job satisfaction. The same feedback sent by email has less of an impact. If the same feedback came from the project manager would be less "authentic". [F]
- Negative project manager progress feedback during a stand up meeting has an impact on behaviour. The same feedback from the division head would be seen as "a bigger problem" [F]
- Positive division head performance emailed feedback has a negative impact on attitude and job satisfaction. The same feedback from the project manager would "feel much more authentic". The same feedback from the CEO would would increase attitude and job satisfaction "in a positive way" [F]
- Negative peer feedback relayed through the project manager has an impact on attitude, behaviour and job satisfaction [F]
- Positive indirect feedback from a peer to a peer or a project manager has an impact on attitude, behaviour, motivation, productivity and job satisfaction [F]
- Negative face to face or emailed peer feedback on breaking the build has an impact on behaviour and productivity [F]
- Positive delayed performance feedback has no impact on the 5 factors [F]

PARTICIPANT 25 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **Yes**

Characteristics

- Source-Technical Respect: No
- Source-Hierarchy: No
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: No
- Source-Credibility: Yes

Demographic [I]

- Developer [I]
- Computer Science Degree [I]
- 5 years experience [I]
- 1 year at Red Gate [I]
- Agile Environments [I]
- Works with 1 other developers in a team of 3 people [I]

Work Desirables [I]

- Doing novel work [I]
- Implementing new features
- Deadlines [I]
- Someone noticing my effort [I]
- Recognition [I]
- Creating something that is [I]used [I]

Work Undesirables [I]

- Working with other peoples code [I]
- Pressure on unsure deadlines [I]
- Overly complicated work [I]
- Unable to fix a problem today [I]

Personality [P]

- Extraversion: **2.65** (MEDIUM)
- Agreeableness: **3.55** (HIGH)
- Conscientiousness: **3.50** (HIGH)
- Openness: **3.55** (HIGH)
- Neuroticism: **2.90** (HIGH)

- Group: **B** (4 people)

Code Impact [I]

- When the code is going right it leads to feeling a sense of completion and achievement[I]
- When code is going wrong it leads to feeling frustrated and under pressure [I]
- When the code is working at the end of the day it leads to feeling satisfied and feeling a sense of achievement [I]
- When the code is not working at the end of the day it leads to feeling annoyed and grumpy, and probably going home [I]

Participant 26 - George

George is a software engineer with 5 years experience after graduating from university with an undergraduate computer science degree. George joined Red Gate 1 year ago with previous experience in agile environments, and now works in a scrum-based agile environment. George works in a small team with one other developer in a team of 3 people.

George enjoys "doing novel work" and "implementing new features". George will go the extra mile if someone notices "I guess it's only worth going the extra mile for me if someone is going to notice I've gone the extra mile" and to get "recognition." George will also go the extra mile to meet deadlines "deadlines are one thing, if something very defiantly has to be done and there will be repercussions if it's not" and George gets satisfaction from making useful products "satisfaction out of producing items of use for society".

George doesn't enjoy working with other peoples code "I don't especially enjoy working with other peoples code" and finds it hard to correct other peoples code "correcting someone elses code which I didn't write so I didn't understand". George doesn't enjoy things he doesn't understand "the thing I dislike most is when things get overly complicated and I don't understand them". When George is unable to fix a task "the only tasks you have to do are where you're mentally not fresh enough to cope with them" he finds this frustrating.

George uses a comparison to targets to assess his progress "it's taking more time than it should is an indication that it's going wrong". George receives feedback from colleagues by their attitude "you can generally tell by their face and their general attitude towards you" and by their body language "by some ones body language". Feedback is received from his manager by their attitude "just pick it up from their general attitude or they'll be more enthusiastic over things". George also receives feedback directly from his manager and colleagues.

George reported that positive face to face peer feedback had an impact on his attitude, behaviour, motivation, productivity and job satisfaction. George would be "more prone to helping people" and it would feel like "a more useful use of my time" and he would "feel I'm being more use". The same feedback emailed would be less "sincere". If the same feedback came from the project manager it would be "a little more removed" and wouldn't have "quite as much" affect. Positive division head performance emailed feedback has an impact on George's attitude and job satisfaction. George would feel "a bit more positive" because "it's nice to feel recognised". The same feedback from the project manager would feel more "genuine". The impact of the same feedback from the CEO would depend on if it's "aligned with something I consider to have done well".

Positive indirect feedback from a peer to a peer or project manager has an impact on George's job satisfaction. George would feel "a bit better about myself" and when people "say good things about me" it "generally makes me feel a bit better" and would make George "feel like I'm being useful". Positive delayed performance feedback has an impact on George's attitude, behaviour and job satisfaction.

Negative project manager or division head progress feedback during a stand up meeting has an impact on George's behaviour and job satisfaction. George would "focus more" on the problem, and he is less satisfied when "somethings not going right". George reported that negative peer feedback relayed through the project manager has an impact on his behaviour. George reported that he would "not do what was causing the problem". George reported that negative face to face or emailed peer feedback on breaking the build has an impact on behaviour. George would try to "not make similar changes in the future" and he would "learn from my mistakes and move on". The same feedback sent by email would have a varying impact depending on who it was also sent to, and if sent to people who "probably didn't need to know" George might feel "a little bit resentful".

Neutral feedback about project progress can have a negative impact on George and make him feel annoyed, frustrated and disappointed. Positive performance feedback from George's manager can have a positive impact and make him feel happy and comforted.

When George is coding and things are going well he feels a "sense of completion" and that he has "achieved something that day". When he is coding and things are not going well George will typically feel "frustrated" and may feel "under pressure" if he doesn't understand what is going on.

At the end of the day when the code works George feels "satisfied" and like he has "achieved something", but if the code doesn't work George will feel "really pissed off" and is "likely to go home and be grumpy".

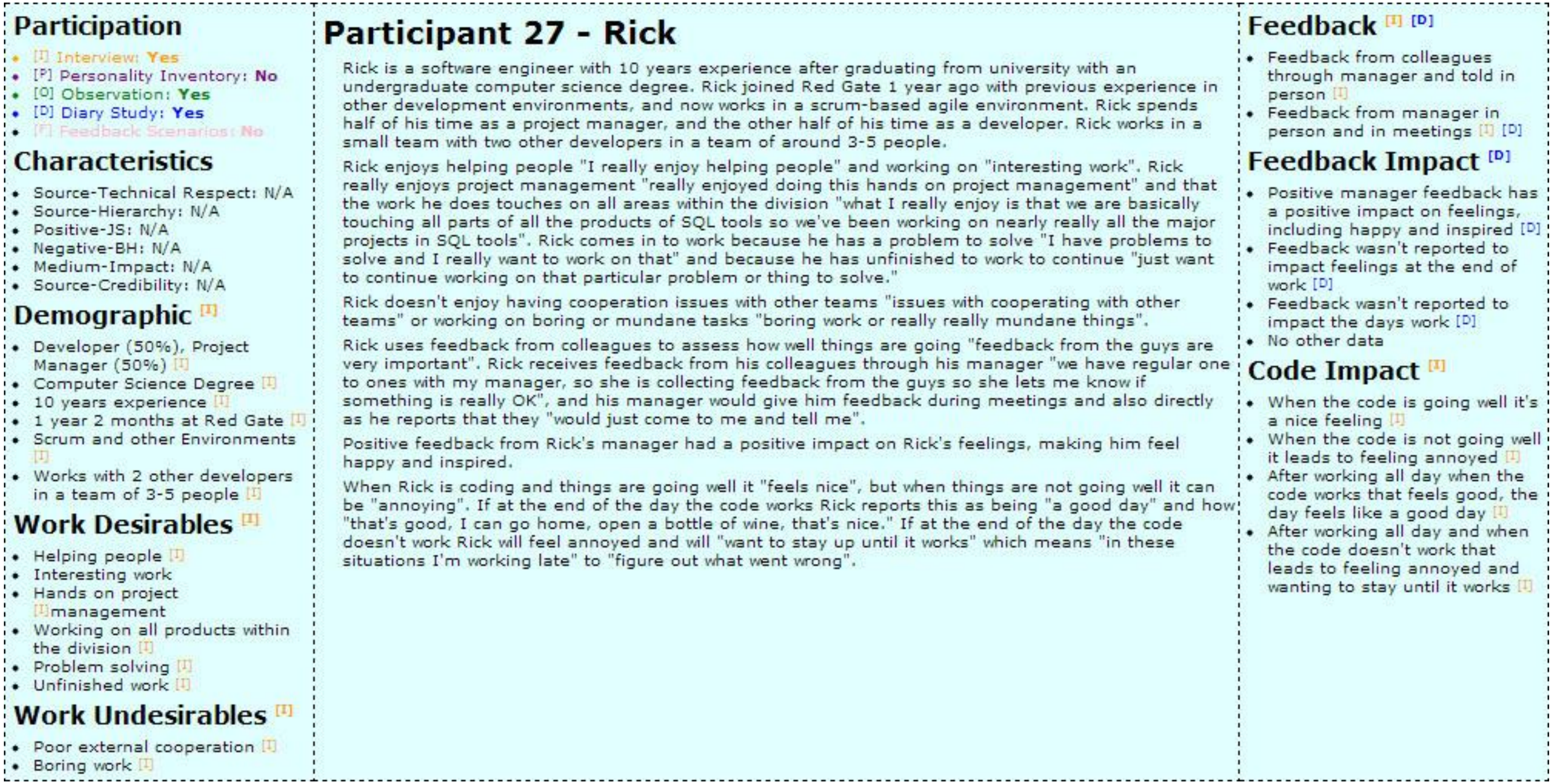
Feedback [I] [D]

- From comparison with targets [I]
- Feedback received from [I]colleagues by their attitude, their body language and their facial expression [I]
- Feedback received from manager by their attitude [I]
- Feedback received from colleagues and managers in person [D]

Feedback Impact [D] [F]

- Feedback can impact feelings [D]
- Neutral feedback can lead to feeling annoyed, frustrated and disappointed [D]
- Positive feedback can lead to feeling happy and comforted [D]
- Feedback wasn't reported to impact feelings at the end of work [D]
- Feedback wasn't reported to impact the days work [D]
- Positive face to face peer feedback has an impact on attitude, behaviour, motivation, productivity and job satisfaction. The same feedback emailed would be less "sincere". If the same feedback came from the project manager it would be "a little more removed [F]
- Negative project manager or division head progress feedback during a stand up meeting has an impact on behaviour and job satisfaction [F]
- Positive division head performance emailed feedback has an impact on attitude and job satisfaction. The same feedback from the project manager would feel more "genuine". The impact of the same feedback from the CEO would vary [F]
- Negative peer feedback relayed through the project manager has an impact on behaviour [F]
- Positive indirect feedback from a peer to a peer or project manager has an impact on job satisfaction [F]
- Negative face to face or emailed peer feedback on breaking the build has an impact on behaviour. The same feedback sent by email would have a varying impact depending on who it was also sent to [F]
- Positive delayed performance feedback has an impact on attitude, behaviour and job satisfaction [F]

PARTICIPANT 26 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **No**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **No**

Characteristics

- Source-Technical Respect: N/A
- Source-Hierarchy: N/A
- Positive-JS: N/A
- Negative-BH: N/A
- Medium-Impact: N/A
- Source-Credibility: N/A

Demographic [I]

- Developer (50%), Project Manager (50%) [I]
- Computer Science Degree [I]
- 10 years experience [I]
- 1 year 2 months at Red Gate [I]
- Scrum and other Environments [I]
- Works with 2 other developers in a team of 3-5 people [I]

Work Desirables [I]

- Helping people [I]
- Interesting work
- Hands on project [I]management
- Working on all products within the division [I]
- Problem solving [I]
- Unfinished work [I]

Work Undesirables [I]

- Poor external cooperation [I]
- Boring work [I]

Participant 27 - Rick

Rick is a software engineer with 10 years experience after graduating from university with an undergraduate computer science degree. Rick joined Red Gate 1 year ago with previous experience in other development environments, and now works in a scrum-based agile environment. Rick spends half of his time as a project manager, and the other half of his time as a developer. Rick works in a small team with two other developers in a team of around 3-5 people.

Rick enjoys helping people "I really enjoy helping people" and working on "interesting work". Rick really enjoys project management "really enjoyed doing this hands on project management" and that the work he does touches on all areas within the division "what I really enjoy is that we are basically touching all parts of all the products of SQL tools so we've been working on nearly really all the major projects in SQL tools". Rick comes in to work because he has a problem to solve "I have problems to solve and I really want to work on that" and because he has unfinished to work to continue "just want to continue working on that particular problem or thing to solve."

Rick doesn't enjoy having cooperation issues with other teams "issues with cooperating with other teams" or working on boring or mundane tasks "boring work or really really mundane things".

Rick uses feedback from colleagues to assess how well things are going "feedback from the guys are very important". Rick receives feedback from his colleagues through his manager "we have regular one to ones with my manager, so she is collecting feedback from the guys so she lets me know if something is really OK", and his manager would give him feedback during meetings and also directly as he reports that they "would just come to me and tell me".

Positive feedback from Rick's manager had a positive impact on Rick's feelings, making him feel happy and inspired.

When Rick is coding and things are going well it "feels nice", but when things are not going well it can be "annoying". If at the end of the day the code works Rick reports this as being "a good day" and how "that's good, I can go home, open a bottle of wine, that's nice." If at the end of the day the code doesn't work Rick will feel annoyed and will "want to stay up until it works" which means "in these situations I'm working late" to "figure out what went wrong".

Feedback [I] [D]

- Feedback from colleagues through manager and told in person [I]
- Feedback from manager in person and in meetings [I] [D]

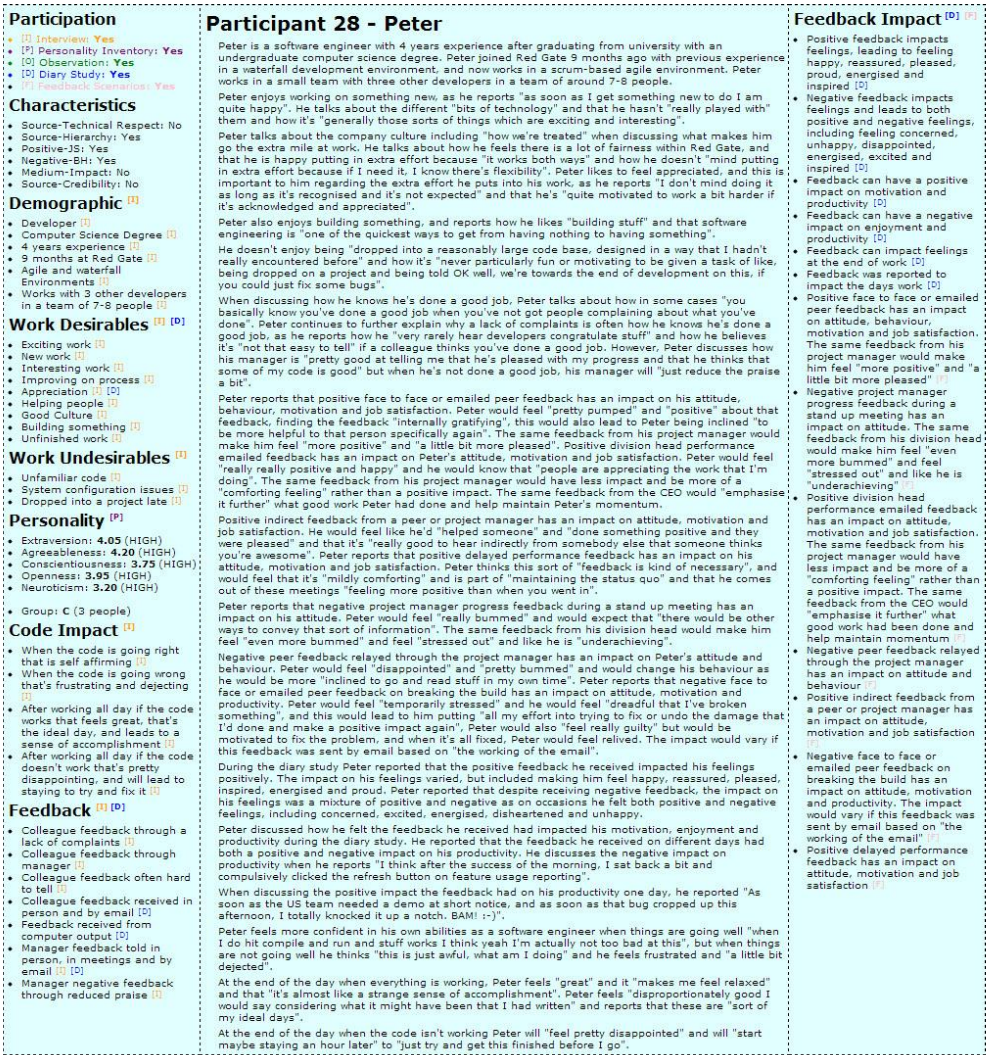
Feedback Impact [D]

- Positive manager feedback has a positive impact on feelings, including happy and inspired [D]
- Feedback wasn't reported to impact feelings at the end of work [D]
- Feedback wasn't reported to impact the days work [D]
- No other data

Code Impact [I]

- When the code is going well it's a nice feeling [I]
- When the code is not going well it leads to feeling annoyed [I]
- After working all day when the code works that feels good, the day feels like a good day [I]
- After working all day and when the code doesn't work that leads to feeling annoyed and wanting to stay until it works [I]

PARTICIPANT 27 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **Yes**
- [O] Observation: **Yes**
- [D] Diary Study: **Yes**
- [F] Feedback Scenarios: **Yes**

Characteristics

- Source-Technical Respect: No
- Source-Hierarchy: Yes
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: No
- Source-Credibility: No

Demographic [I]

- Developer [I]
- Computer Science Degree [I]
- 4 years experience [I]
- 9 months at Red Gate [I]
- Agile and waterfall Environments [I]
- Works with 3 other developers in a team of 7-8 people [I]

Work Desirables [I] [D]

- Exciting work [I]
- New work [I]
- Interesting work [I]
- Improving on process [I]
- Appreciation [I] [D]
- Helping people [I]
- Good Culture [I]
- Building something [I]
- Unfinished work [I]

Work Undesirables [I]

- Unfamiliar code [I]
- System configuration issues [I]
- Dropped into a project late [I]

Personality [P]

- Extraversion: **4.05** (HIGH)
- Agreeableness: **4.20** (HIGH)
- Conscientiousness: **3.75** (HIGH)
- Openness: **3.95** (HIGH)
- Neuroticism: **3.20** (HIGH)

- Group: **C** (3 people)

Code Impact [I]

- When the code is going right that is self affirming [I]
- When the code is going wrong that's frustrating and dejecting [I]
- After working all day if the code works that feels great, that's the ideal day, and leads to a sense of accomplishment [I]
- After working all day if the code doesn't work that's pretty disappointing, and will lead to staying to try and fix it [I]

Feedback [I] [D]

- Colleague feedback through a lack of complaints [I]
- Colleague feedback through manager [I]
- Colleague feedback often hard to tell [I]
- Colleague feedback received in person and by email [D]
- Feedback received from computer output [D]
- Manager feedback told in person, in meetings and by email [I] [D]
- Manager negative feedback through reduced praise [I]

Participant 28 - Peter

Peter is a software engineer with 4 years experience after graduating from university with an undergraduate computer science degree. Peter joined Red Gate 9 months ago with previous experience in a waterfall development environment, and now works in a scrum-based agile environment. Peter works in a small team with three other developers in a team of around 7-8 people.

Peter enjoys working on something new, as he reports "as soon as I get something new to do I am quite happy". He talks about the different "bits of technology" and that he hasn't "really played with" them and how it's "generally those sorts of things which are exciting and interesting".

Peter talks about the company culture including "how we're treated" when discussing what makes him go the extra mile at work. He talks about how he feels there is a lot of fairness within Red Gate, and that he is happy putting in extra effort because "it works both ways" and how he doesn't "mind putting in extra effort because if I need it, I know there's flexibility". Peter likes to feel appreciated, and this is important to him regarding the extra effort he puts into his work, as he reports "I don't mind doing it as long as it's recognised and it's not expected" and that he's "quite motivated to work a bit harder if it's acknowledged and appreciated".

Peter also enjoys building something, and reports how he likes "building stuff" and that software engineering is "one of the quickest ways to get from having nothing to having something".

He doesn't enjoy being "dropped into a reasonably large code base, designed in a way that I hadn't really encountered before" and how it's "never particularly fun or motivating to be given a task of like, being dropped on a project and being told OK well, we're towards the end of development on this, if you could just fix some bugs".

When discussing how he knows he's done a good job, Peter talks about how in some cases "you basically know you've done a good job when you've not got people complaining about what you've done". Peter continues to further explain why a lack of complaints is often how he knows he's done a good job, as he reports how he "very rarely hear developers congratulate stuff" and how he believes it's "not that easy to tell" if a colleague thinks you've done a good job. However, Peter discusses how his manager is "pretty good at telling me that he's pleased with my progress and that he thinks that some of my code is good" but when he's not done a good job, his manager will "just reduce the praise a bit".

Peter reports that positive face to face or emailed peer feedback has an impact on his attitude, behaviour, motivation and job satisfaction. Peter would feel "pretty pumped" and "positive" about that feedback, finding the feedback "internally gratifying", this would also lead to Peter being inclined "to be more helpful to that person specifically again". The same feedback from his project manager would make him feel "more positive" and "a little bit more pleased". Positive division head performance emailed feedback has an impact on Peter's attitude, motivation and job satisfaction. Peter would feel "really really positive and happy" and he would know that "people are appreciating the work that I'm doing". The same feedback from his project manager would have less impact and be more of a "comforting feeling" rather than a positive impact. The same feedback from the CEO would "emphasise it further" what good work Peter had done and help maintain Peter's momentum.

Positive indirect feedback from a peer or project manager has an impact on attitude, motivation and job satisfaction. He would feel like he'd "helped someone" and "done something positive and they were pleased" and that it's "really good to hear indirectly from somebody else that someone thinks you're awesome". Peter reports that positive delayed performance feedback has an impact on his attitude, motivation and job satisfaction. Peter thinks this sort of "feedback is kind of necessary", and would feel that it's "mildly comforting" and is part of "maintaining the status quo" and that he comes out of these meetings "feeling more positive than when you went in".

Peter reports that negative project manager progress feedback during a stand up meeting has an impact on his attitude. Peter would feel "really bummed" and would expect that "there would be other ways to convey that sort of information". The same feedback from his division head would make him feel "even more bummed" and feel "stressed out" and like he is "underachieving".

Negative peer feedback relayed through the project manager has an impact on Peter's attitude and behaviour. Peter would feel "disappointed" and "pretty bummed" and would change his behaviour as he would be more "inclined to go and read stuff in my own time". Peter reports that negative face to face or emailed peer feedback on breaking the build has an impact on attitude, motivation and productivity. Peter would feel "temporarily stressed" and he would feel "dreadful that I've broken something", and this would lead to him putting "all my effort into trying to fix or undo the damage that I'd done and make a positive impact again", Peter would also "feel really guilty" but would be motivated to fix the problem, and when it's all fixed, Peter would feel relived. The impact would vary if this feedback was sent by email based on "the working of the email".

During the diary study Peter reported that the positive feedback he received impacted his feelings positively. The impact on his feelings varied, but included making him feel happy, reassured, pleased, inspired, energised and proud. Peter reported that despite receiving negative feedback, the impact on his feelings was a mixture of positive and negative as on occasions he felt both positive and negative feelings, including concerned, excited, energised, disheartened and unhappy.

Peter discussed how he felt the feedback he received had impacted his motivation, enjoyment and productivity during the diary study. He reported that the feedback he received on different days had both a positive and negative impact on his productivity. He discusses the negative impact on productivity when he reports "I think after the success of the morning, I sat back a bit and compulsively clicked the refresh button on feature usage reporting".

When discussing the positive impact the feedback had on his productivity one day, he reported "As soon as the US team needed a demo at short notice, and as soon as that bug cropped up this afternoon, I totally knocked it up a notch. BAM! :-)".

Peter feels more confident in his own abilities as a software engineer when things are going well "when I do hit compile and run and stuff works I think yeah I'm actually not too bad at this", but when things are not going well he thinks "this is just awful, what am I doing" and he feels frustrated and "a little bit dejected".

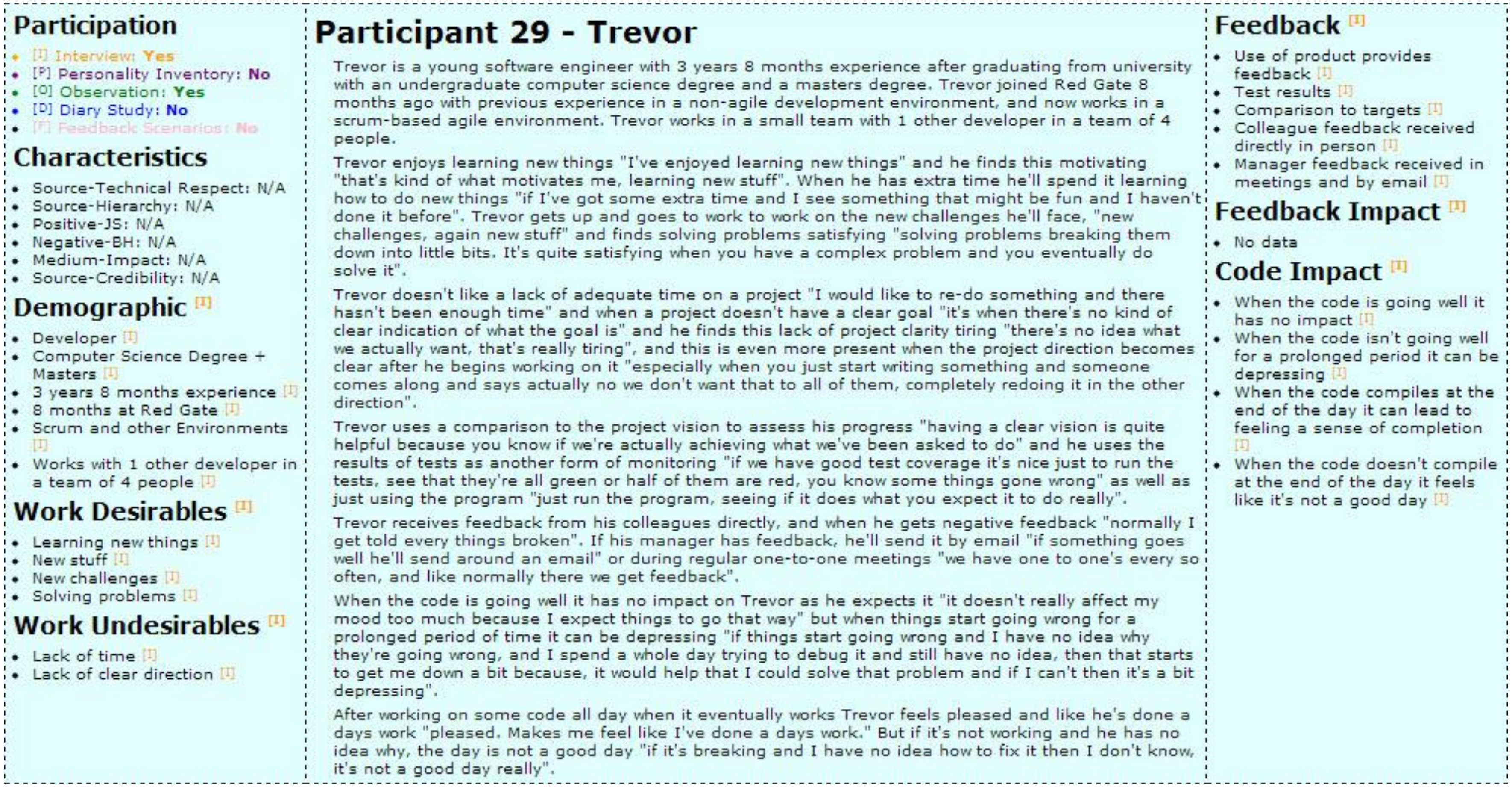
At the end of the day when everything is working, Peter feels "great" and it "makes me feel relaxed" and that "it's almost like a strange sense of accomplishment". Peter feels "disproportionately good I would say considering what it might have been that I had written" and reports that these are "sort of my ideal days".

At the end of the day when the code isn't working Peter will "feel pretty disappointed" and will "start maybe staying an hour later" to "just try and get this finished before I go".

Feedback Impact [D] [F]

- Positive feedback impacts feelings, leading to feeling happy, reassured, pleased, proud, energised and inspired [D]
- Negative feedback impacts feelings and leads to both positive and negative feelings, including feeling concerned, unhappy, disappointed, energised, excited and inspired [D]
- Feedback can have a positive impact on motivation and productivity [D]
- Feedback can have a negative impact on enjoyment and productivity [D]
- Feedback can impact feelings at the end of work [D]
- Feedback was reported to impact the days work [D]
- Positive face to face or emailed peer feedback has an impact on attitude, behaviour, motivation and job satisfaction. The same feedback from his project manager would make him feel "more positive" and "a little bit more pleased" [F]
- Negative project manager progress feedback during a stand up meeting has an impact on attitude. The same feedback from his division head would make him feel "even more bummed" and feel "stressed out" and like he is "underachieving" [F]
- Positive division head performance emailed feedback has an impact on attitude, motivation and job satisfaction. The same feedback from his project manager would have less impact and be more of a "comforting feeling" rather than a positive impact. The same feedback from the CEO would "emphasise it further" what good work had been done and help maintain momentum [F]
- Negative peer feedback relayed through the project manager has an impact on attitude and behaviour [F]
- Positive indirect feedback from a peer or project manager has an impact on attitude, motivation and job satisfaction [F]
- Negative face to face or emailed peer feedback on breaking the build has an impact on attitude, motivation and productivity. The impact would vary if this feedback was sent by email based on "the working of the email" [F]
- Positive delayed performance feedback has an impact on attitude, motivation and job satisfaction [F]

PARTICIPANT 28 PERSONAL PROFILE



Participation

- [I] Interview: **Yes**
- [P] Personality Inventory: **No**
- [O] Observation: **Yes**
- [D] Diary Study: **No**
- [F] Feedback Scenarios: **No**

Characteristics

- Source-Technical Respect: N/A
- Source-Hierarchy: N/A
- Positive-JS: N/A
- Negative-BH: N/A
- Medium-Impact: N/A
- Source-Credibility: N/A

Demographic [I]

- Developer [I]
- Computer Science Degree + Masters [I]
- 3 years 8 months experience [I]
- 8 months at Red Gate [I]
- Scrum and other Environments [I]
- Works with 1 other developer in a team of 4 people [I]

Work Desirables [I]

- Learning new things [I]
- New stuff [I]
- New challenges [I]
- Solving problems [I]

Work Undesirables [I]

- Lack of time [I]
- Lack of clear direction [I]

Participant 29 - Trevor

Trevor is a young software engineer with 3 years 8 months experience after graduating from university with an undergraduate computer science degree and a masters degree. Trevor joined Red Gate 8 months ago with previous experience in a non-agile development environment, and now works in a scrum-based agile environment. Trevor works in a small team with 1 other developer in a team of 4 people.

Trevor enjoys learning new things "I've enjoyed learning new things" and he finds this motivating "that's kind of what motivates me, learning new stuff". When he has extra time he'll spend it learning how to do new things "if I've got some extra time and I see something that might be fun and I haven't done it before". Trevor gets up and goes to work to work on the new challenges he'll face, "new challenges, again new stuff" and finds solving problems satisfying "solving problems breaking them down into little bits. It's quite satisfying when you have a complex problem and you eventually do solve it".

Trevor doesn't like a lack of adequate time on a project "I would like to re-do something and there hasn't been enough time" and when a project doesn't have a clear goal "it's when there's no kind of clear indication of what the goal is" and he finds this lack of project clarity tiring "there's no idea what we actually want, that's really tiring", and this is even more present when the project direction becomes clear after he begins working on it "especially when you just start writing something and someone comes along and says actually no we don't want that to all of them, completely redoing it in the other direction".

Trevor uses a comparison to the project vision to assess his progress "having a clear vision is quite helpful because you know if we're actually achieving what we've been asked to do" and he uses the results of tests as another form of monitoring "if we have good test coverage it's nice just to run the tests, see that they're all green or half of them are red, you know some things gone wrong" as well as just using the program "just run the program, seeing if it does what you expect it to do really".

Trevor receives feedback from his colleagues directly, and when he gets negative feedback "normally I get told every things broken". If his manager has feedback, he'll send it by email "if something goes well he'll send around an email" or during regular one-to-one meetings "we have one to one's every so often, and like normally there we get feedback".

When the code is going well it has no impact on Trevor as he expects it "it doesn't really affect my mood too much because I expect things to go that way" but when things start going wrong for a prolonged period of time it can be depressing "if things start going wrong and I have no idea why they're going wrong, and I spend a whole day trying to debug it and still have no idea, then that starts to get me down a bit because, it would help that I could solve that problem and if I can't then it's a bit depressing".

After working on some code all day when it eventually works Trevor feels pleased and like he's done a days work "pleased. Makes me feel like I've done a days work." But if it's not working and he has no idea why, the day is not a good day "if it's breaking and I have no idea how to fix it then I don't know, it's not a good day really".

Feedback [I]

- Use of product provides feedback [I]
- Test results [I]
- Comparison to targets [I]
- Colleague feedback received directly in person [I]
- Manager feedback received in meetings and by email [I]

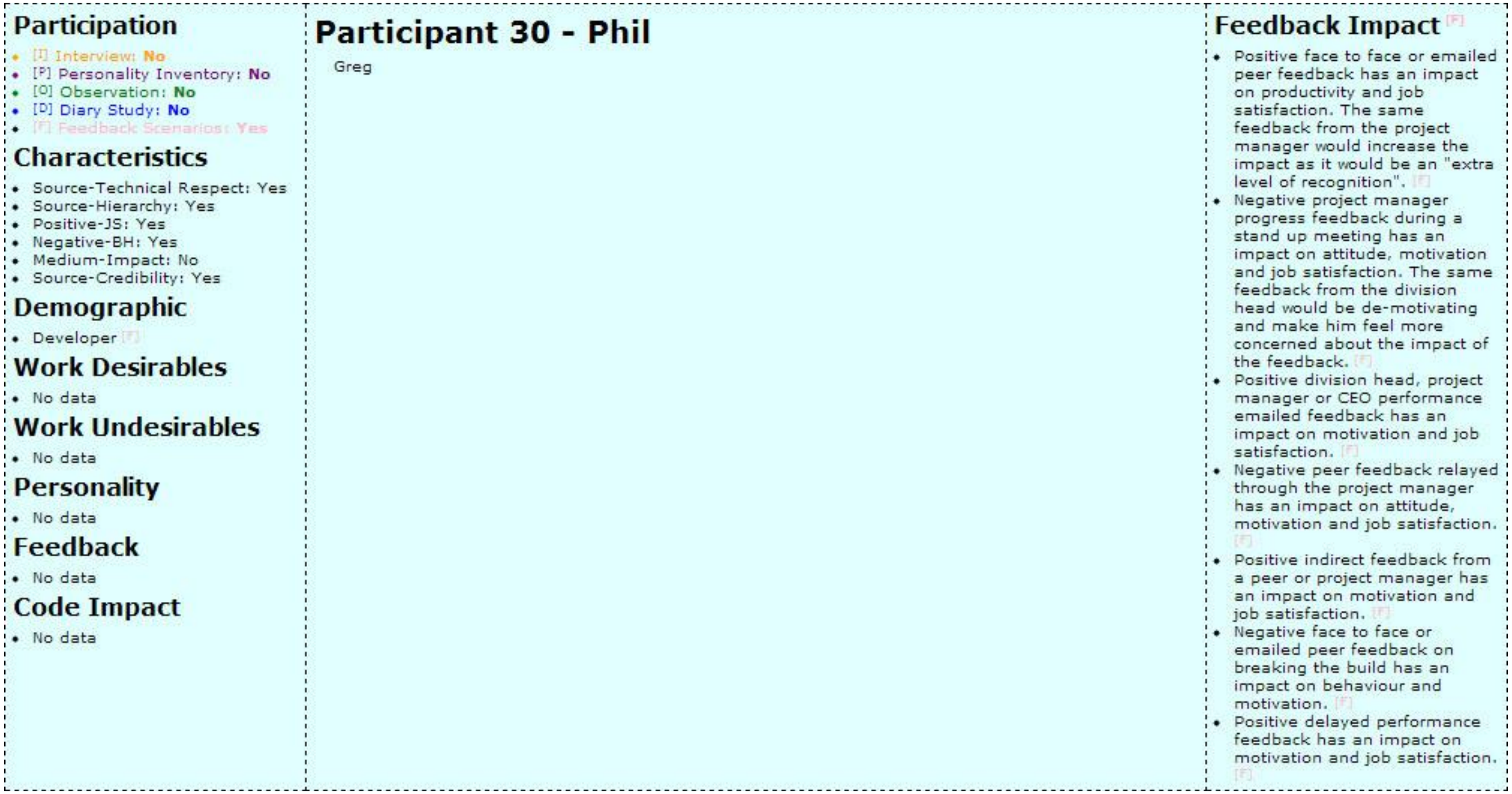
Feedback Impact [I]

- No data

Code Impact [I]

- When the code is going well it has no impact [I]
- When the code isn't going well for a prolonged period it can be depressing [I]
- When the code compiles at the end of the day it can lead to feeling a sense of completion [I]
- When the code doesn't compile at the end of the day it feels like it's not a good day [I]

PARTICIPANT 29 PERSONAL PROFILE



Participation

- [I] Interview: **No**
- [P] Personality Inventory: **No**
- [O] Observation: **No**
- [D] Diary Study: **No**
- [F] Feedback Scenarios: **Yes**

Characteristics

- Source-Technical Respect: Yes
- Source-Hierarchy: Yes
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: No
- Source-Credibility: Yes

Demographic

- Developer [F]

Work Desirables

- No data

Work Undesirables

- No data

Personality

- No data

Feedback

- No data

Code Impact

- No data

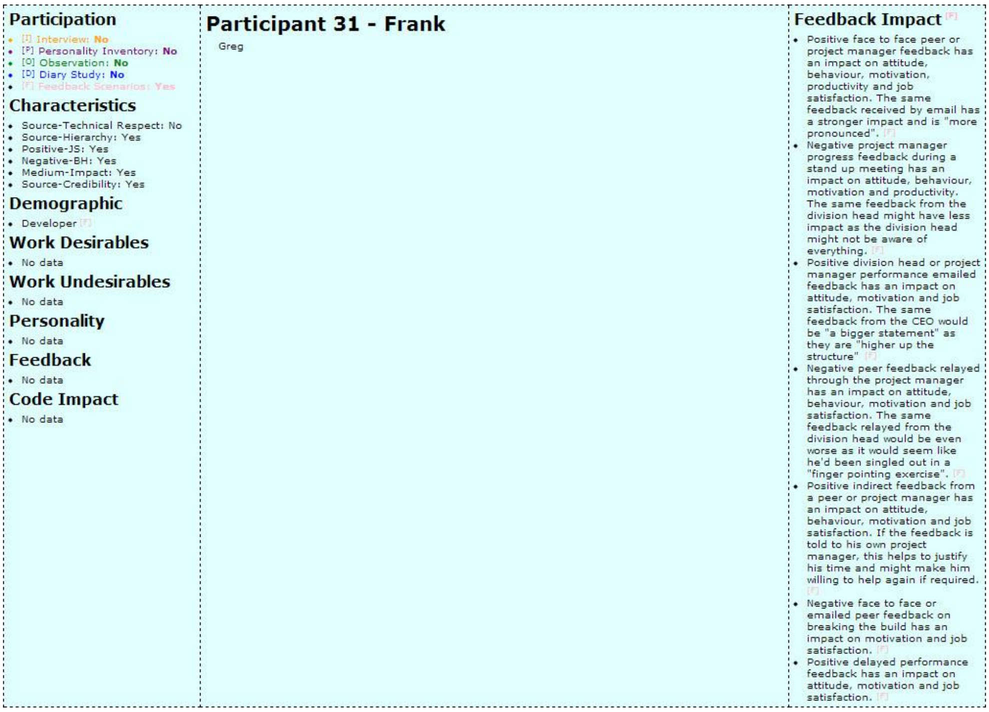
Participant 30 - Phil

Greg

Feedback Impact [F]

- Positive face to face or emailed peer feedback has an impact on productivity and job satisfaction. The same feedback from the project manager would increase the impact as it would be an "extra level of recognition". [F]
- Negative project manager progress feedback during a stand up meeting has an impact on attitude, motivation and job satisfaction. The same feedback from the division head would be de-motivating and make him feel more concerned about the impact of the feedback. [F]
- Positive division head, project manager or CEO performance emailed feedback has an impact on motivation and job satisfaction. [F]
- Negative peer feedback relayed through the project manager has an impact on attitude, motivation and job satisfaction. [F]
- Positive indirect feedback from a peer or project manager has an impact on motivation and job satisfaction. [F]
- Negative face to face or emailed peer feedback on breaking the build has an impact on behaviour and motivation. [F]
- Positive delayed performance feedback has an impact on motivation and job satisfaction. [F]

PARTICIPANT 30 PERSONAL PROFILE



Participation

- [I] Interview: No
- [P] Personality Inventory: No
- [O] Observation: No
- [D] Diary Study: No
- [F] Feedback Scenarios: Yes

Characteristics

- Source-Technical Respect: No
- Source-Hierarchy: Yes
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: Yes
- Source-Credibility: Yes

Demographic

- Developer [F]

Work Desirables

- No data

Work Undesirables

- No data

Personality

- No data

Feedback

- No data

Code Impact

- No data

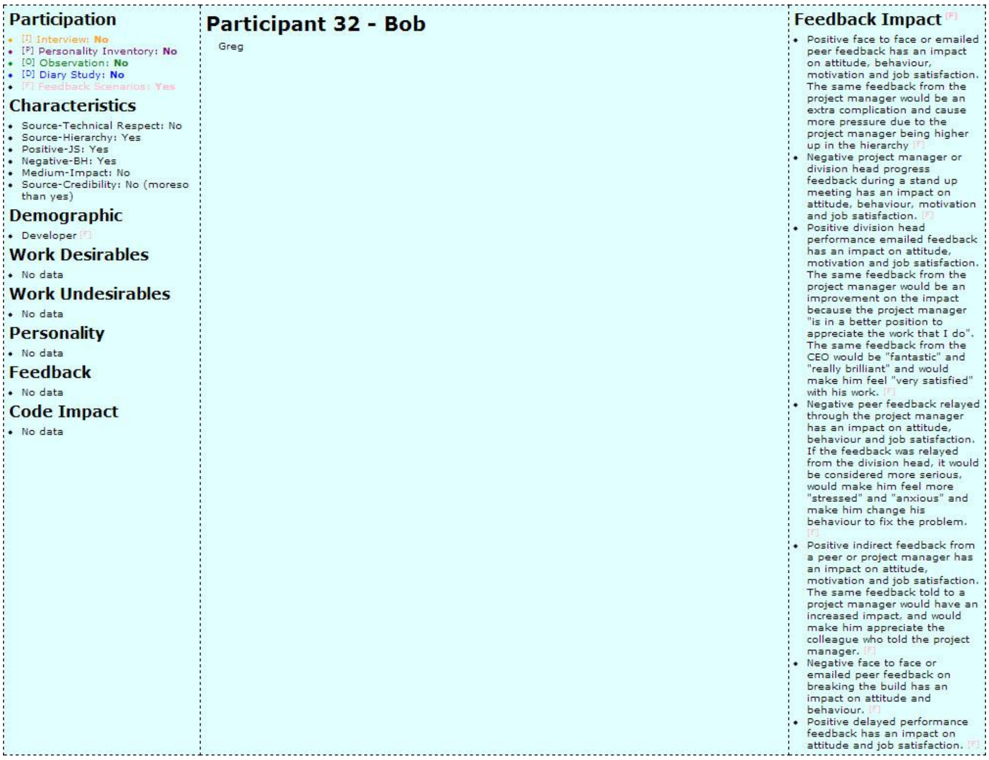
Participant 31 - Frank

Greg

Feedback Impact [F]

- Positive face to face peer or project manager feedback has an impact on attitude, behaviour, motivation, productivity and job satisfaction. The same feedback received by email has a stronger impact and is "more pronounced". [F]
- Negative project manager progress feedback during a stand up meeting has an impact on attitude, behaviour, motivation and productivity. The same feedback from the division head might have less impact as the division head might not be aware of everything. [F]
- Positive division head or project manager performance emailed feedback has an impact on attitude, motivation and job satisfaction. The same feedback from the CEO would be "a bigger statement" as they are "higher up the structure" [F]
- Negative peer feedback relayed through the project manager has an impact on attitude, behaviour, motivation and job satisfaction. The same feedback relayed from the division head would be even worse as it would seem like he'd been singled out in a "finger pointing exercise". [F]
- Positive indirect feedback from a peer or project manager has an impact on attitude, behaviour, motivation and job satisfaction. If the feedback is told to his own project manager, this helps to justify his time and might make him willing to help again if required. [F]
- Negative face to face or emailed peer feedback on breaking the build has an impact on motivation and job satisfaction. [F]
- Positive delayed performance feedback has an impact on attitude, motivation and job satisfaction. [F]

PARTICIPANT 31 PERSONAL PROFILE



Participation

- [I] Interview: **No**
- [P] Personality Inventory: **No**
- [O] Observation: **No**
- [D] Diary Study: **No**
- [F] Feedback Scenarios: **Yes**

Characteristics

- Source-Technical Respect: No
- Source-Hierarchy: Yes
- Positive-JS: Yes
- Negative-BH: Yes
- Medium-Impact: No
- Source-Credibility: No (moreso than yes)

Demographic

- Developer [F]

Work Desirables

- No data

Work Undesirables

- No data

Personality

- No data

Feedback

- No data

Code Impact

- No data

Participant 32 - Bob

Greg

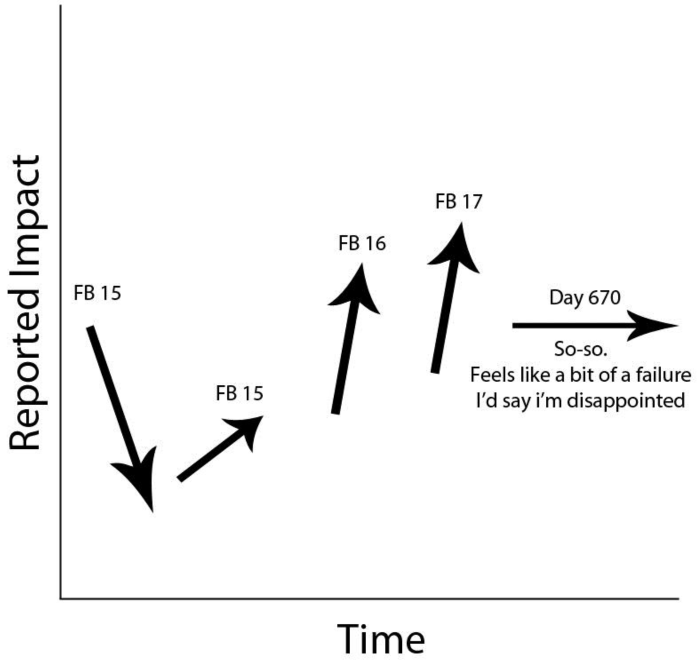
Feedback Impact [F]

- Positive face to face or emailed peer feedback has an impact on attitude, behaviour, motivation and job satisfaction. The same feedback from the project manager would be an extra complication and cause more pressure due to the project manager being higher up in the hierarchy [F]
- Negative project manager or division head progress feedback during a stand up meeting has an impact on attitude, behaviour, motivation and job satisfaction. [F]
- Positive division head performance emailed feedback has an impact on attitude, motivation and job satisfaction. The same feedback from the project manager would be an improvement on the impact because the project manager "is in a better position to appreciate the work that I do". The same feedback from the CEO would be "fantastic" and "really brilliant" and would make him feel "very satisfied" with his work. [F]
- Negative peer feedback relayed through the project manager has an impact on attitude, behaviour and job satisfaction. If the feedback was relayed from the division head, it would be considered more serious, would make him feel more "stressed" and "anxious" and make him change his behaviour to fix the problem. [F]
- Positive indirect feedback from a peer or project manager has an impact on attitude, motivation and job satisfaction. The same feedback told to a project manager would have an increased impact, and would make him appreciate the colleague who told the project manager. [F]
- Negative face to face or emailed peer feedback on breaking the build has an impact on attitude and behaviour. [F]
- Positive delayed performance feedback has an impact on attitude and job satisfaction. [F]

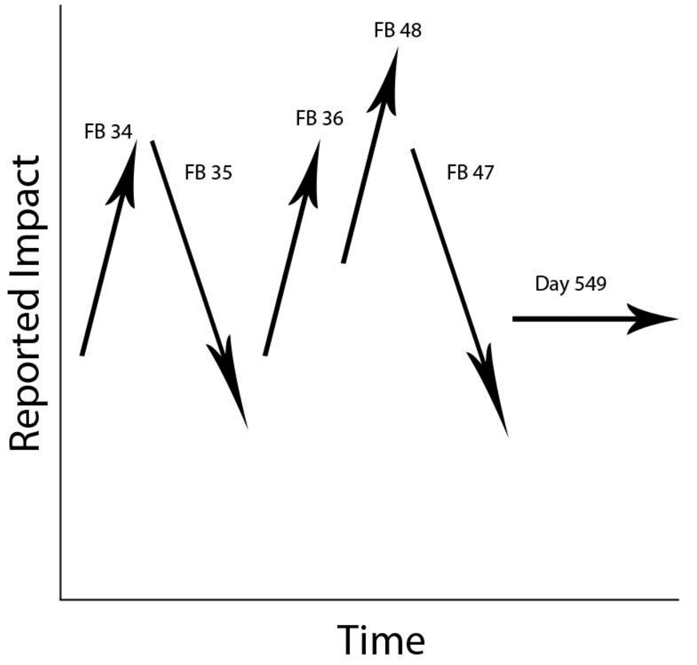
PARTICIPANT 32 PERSONAL PROFILE

APPENDIX B SECTION 6 – FEEDBACK IN ONE SOFTWARE ENGINEERING ENVIRONMENT STUDY DAY AND WEEK PATH ANALYSIS FIGURES

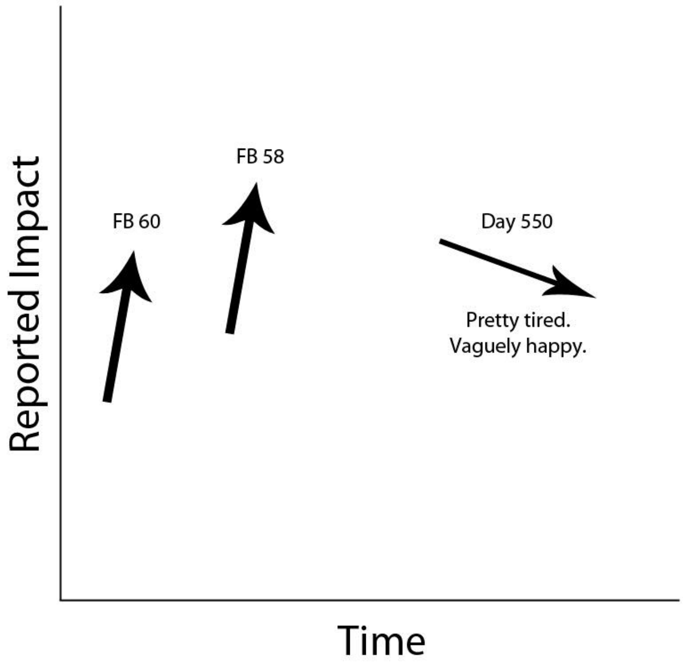
DAY PATH ANALYSIS



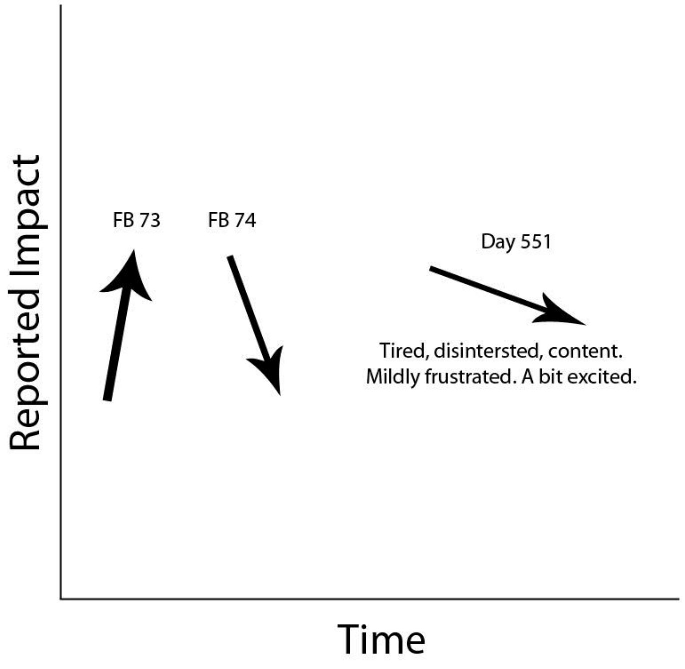
DAY PATH ANALYSIS FOR DAY 548



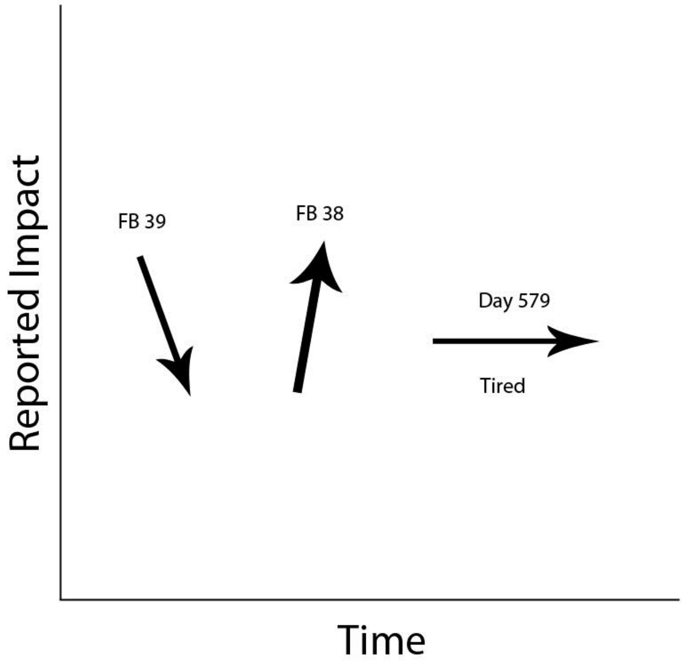
DAY PATH ANALYSIS FOR DAY 549



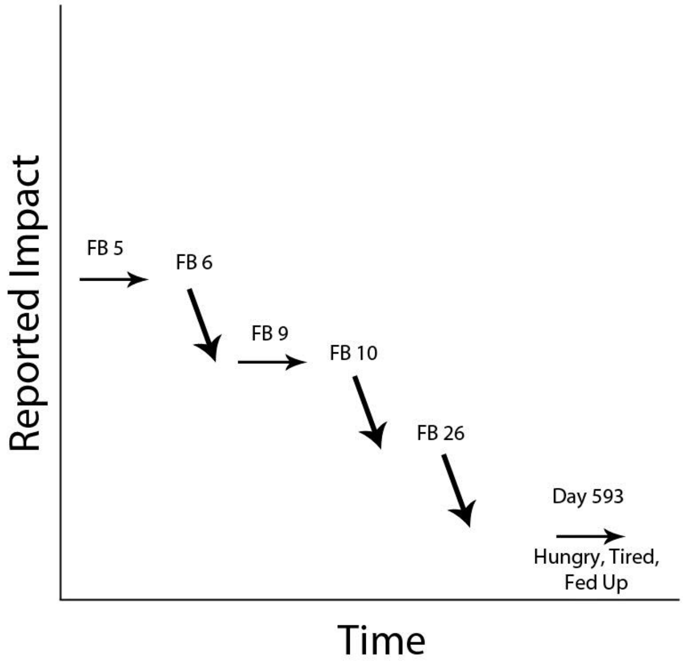
DAY PATH ANALYSIS FOR DAY 550


Reported Impact
FB 73
FB 74
Day 551
Tired, disintersted, content.
Mildly frustrated. A bit excited.
Time

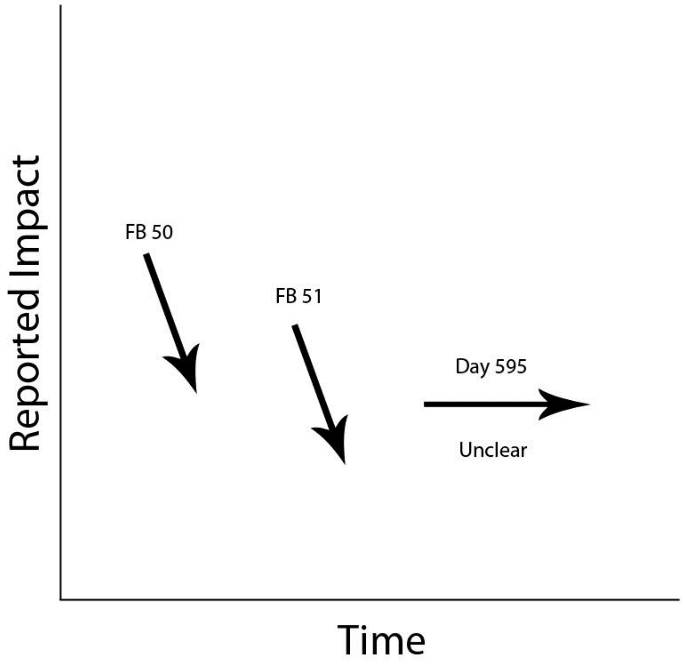
DAY PATH ANALYSIS FOR DAY 551



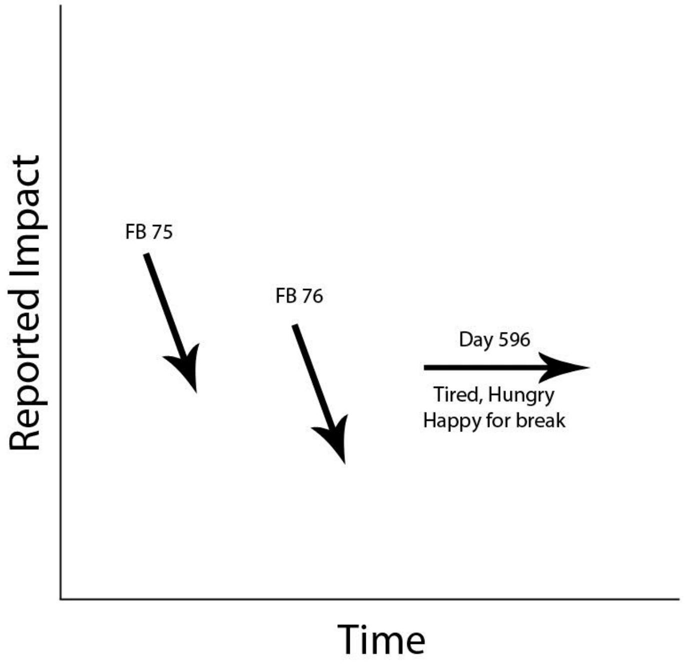
DAY PATH ANALYSIS FOR DAY 579



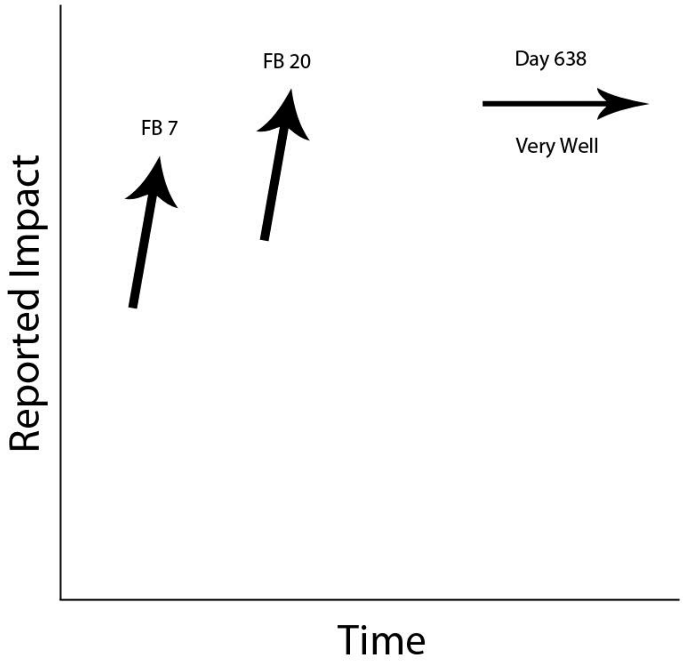
DAY PATH ANALYSIS FOR DAY 593



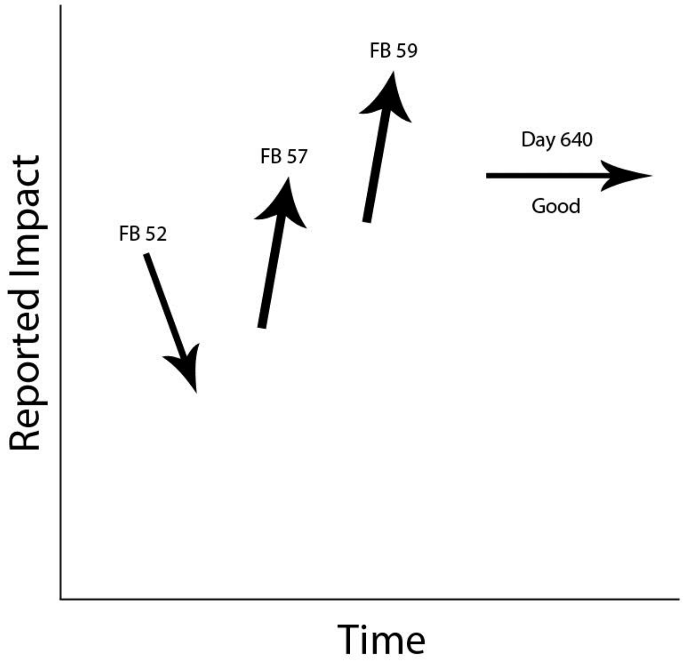
DAY PATH ANALYSIS FOR DAY 595



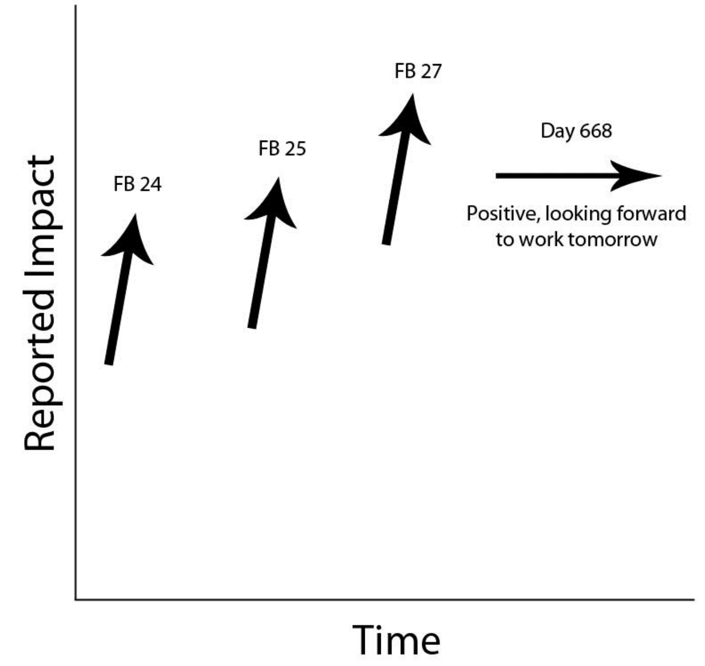
DAY PATH ANALYSIS FOR DAY 596



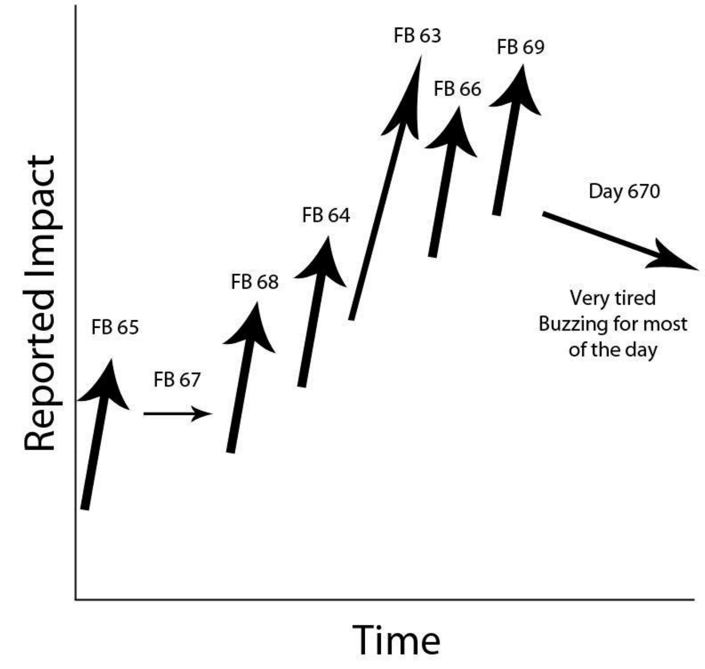
DAY PATH ANALYSIS FOR DAY 638



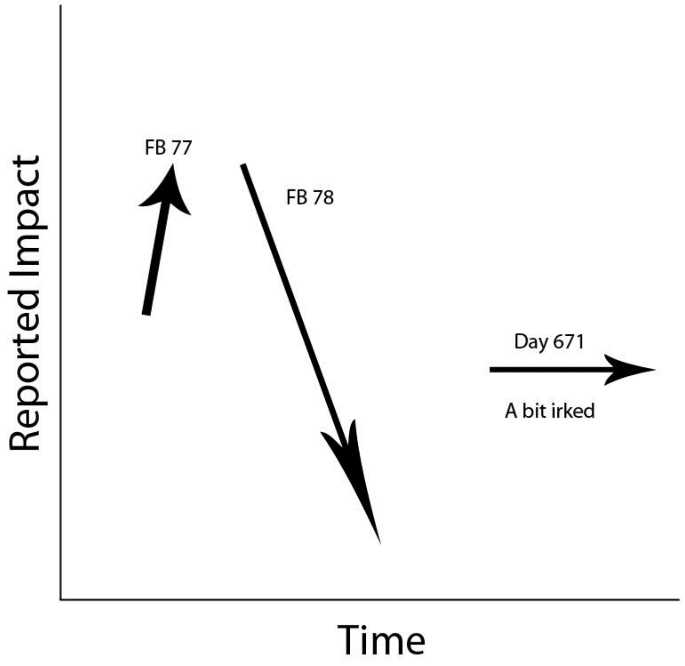
DAY PATH ANALYSIS FOR DAY 640



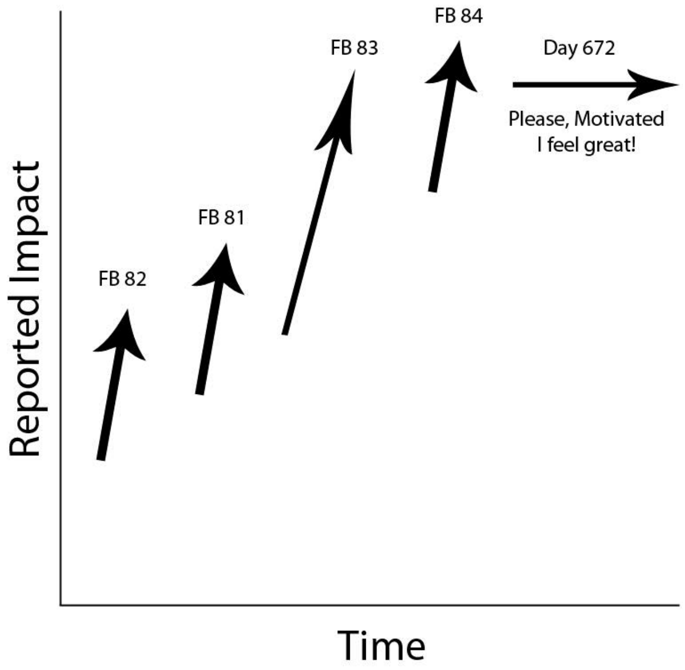
DAY PATH ANALYSIS FOR DAY 668



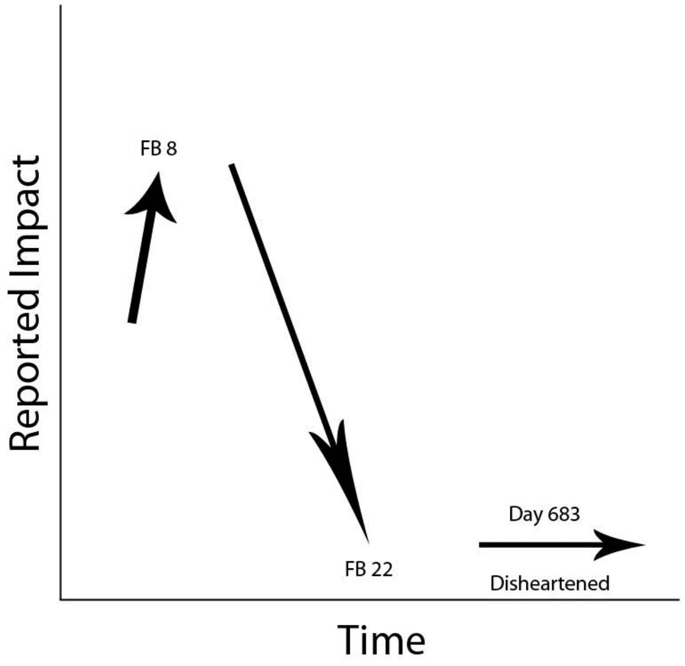
DAY PATH ANALYSIS FOR DAY 670



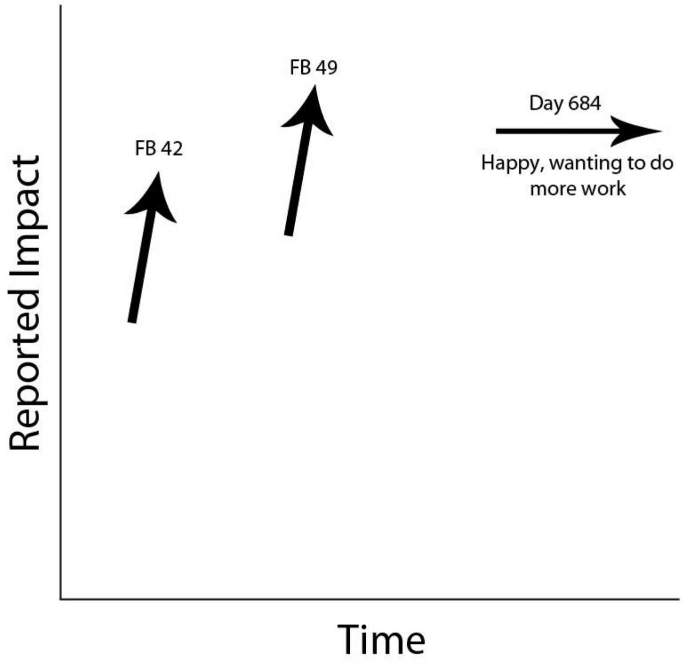
DAY PATH ANALYSIS FOR DAY 671



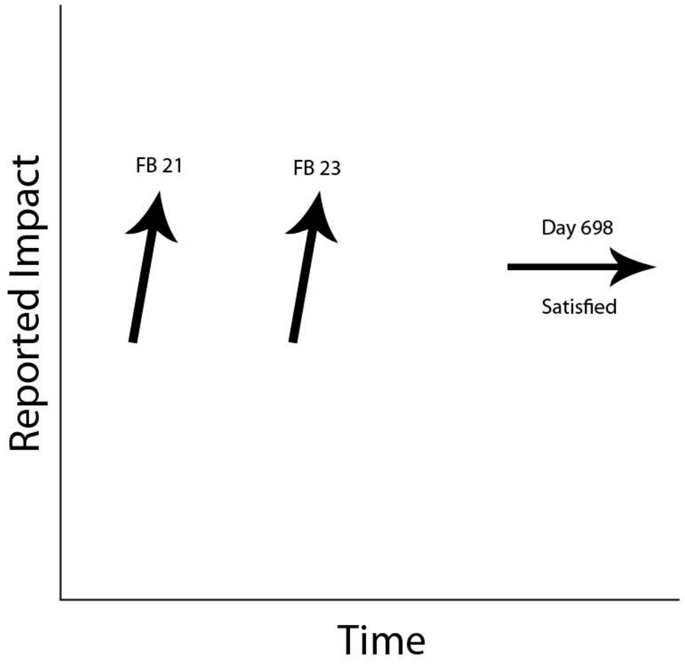
DAY PATH ANALYSIS FOR DAY 672



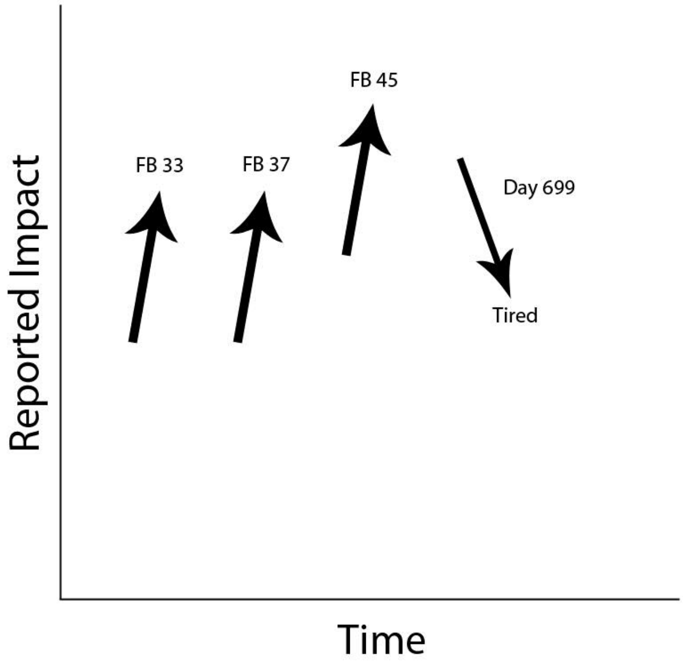
DAY PATH ANALYSIS FOR DAY 683



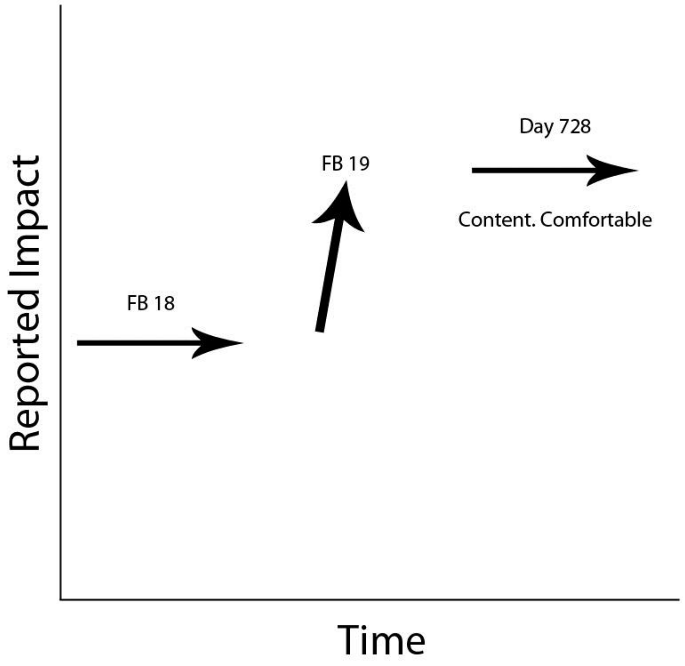
DAY PATH ANALYSIS FOR DAY 684



DAY PATH ANALYSIS FOR DAY 699

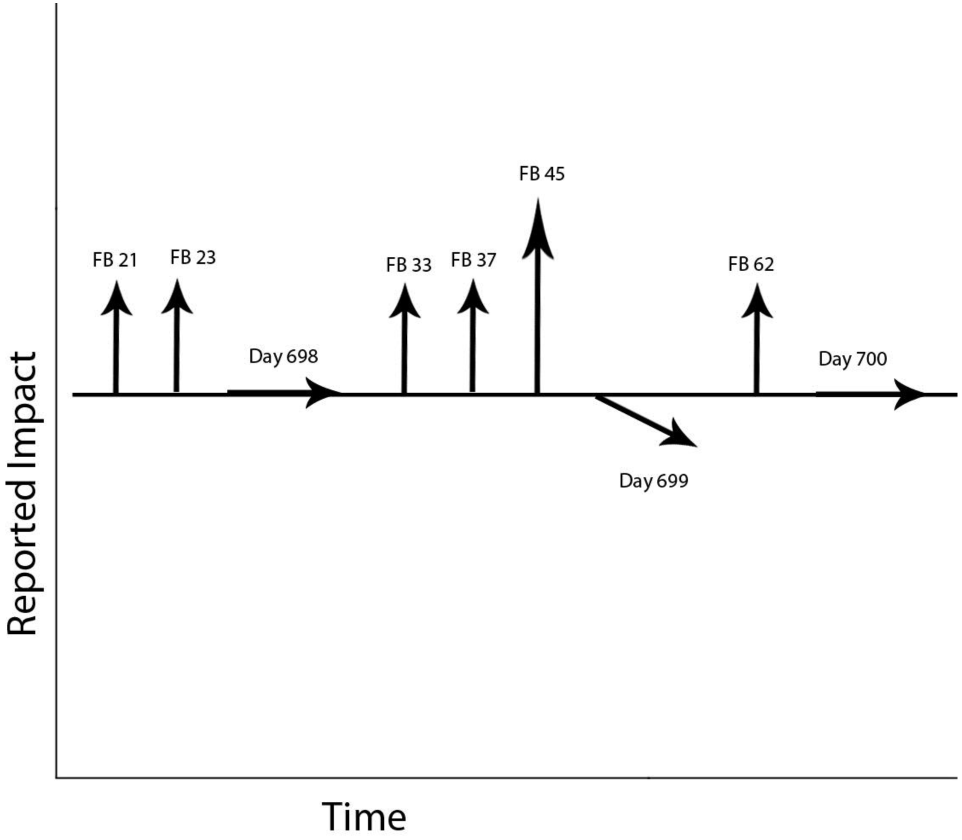


DAY PATH ANALYSIS FOR DAY 699

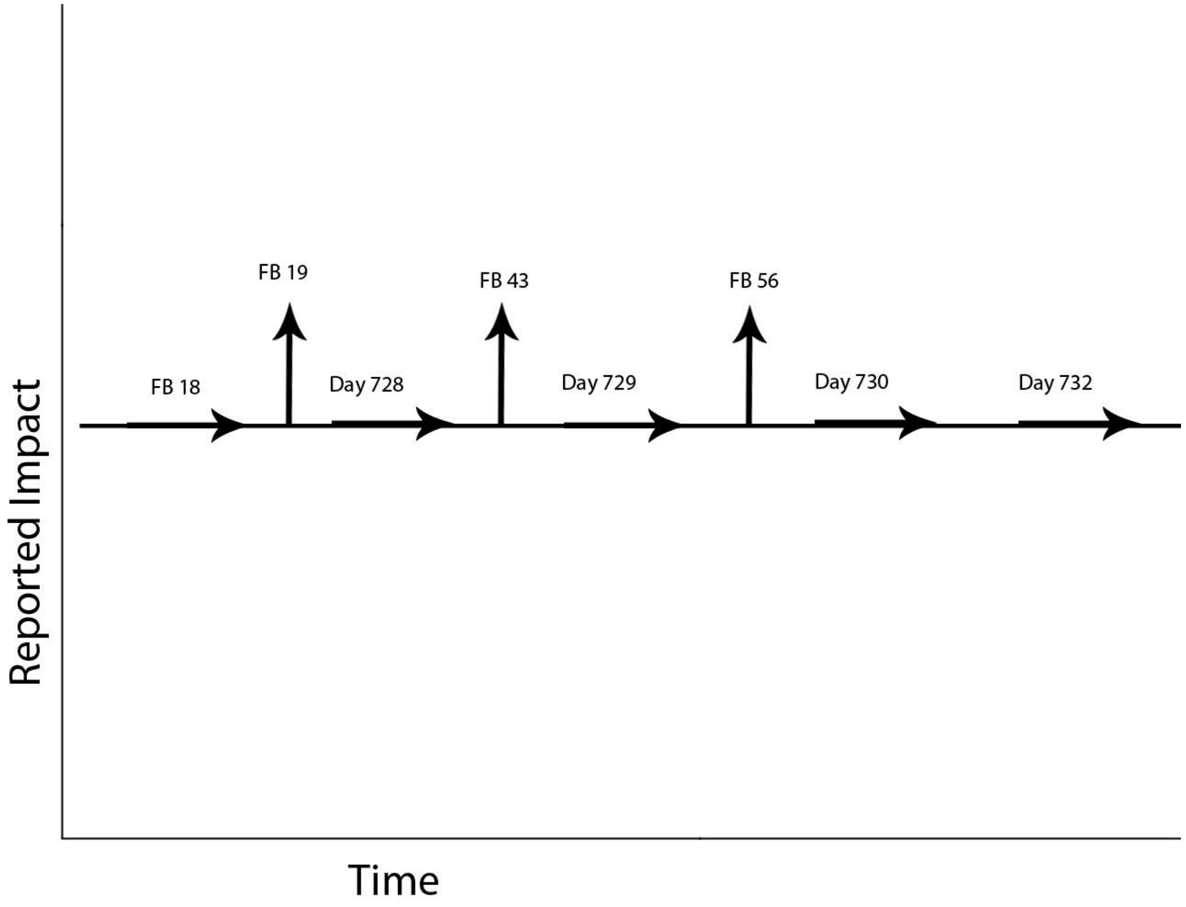


DAY PATH ANALYSIS FOR DAY 728

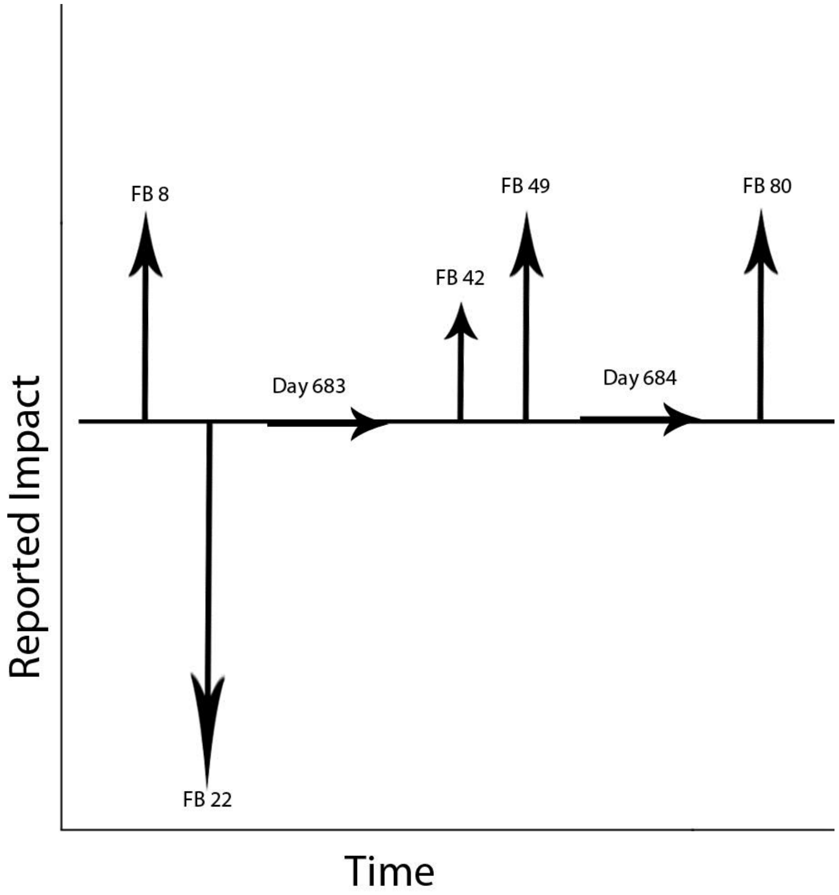
WEEK PATH ANALYSIS



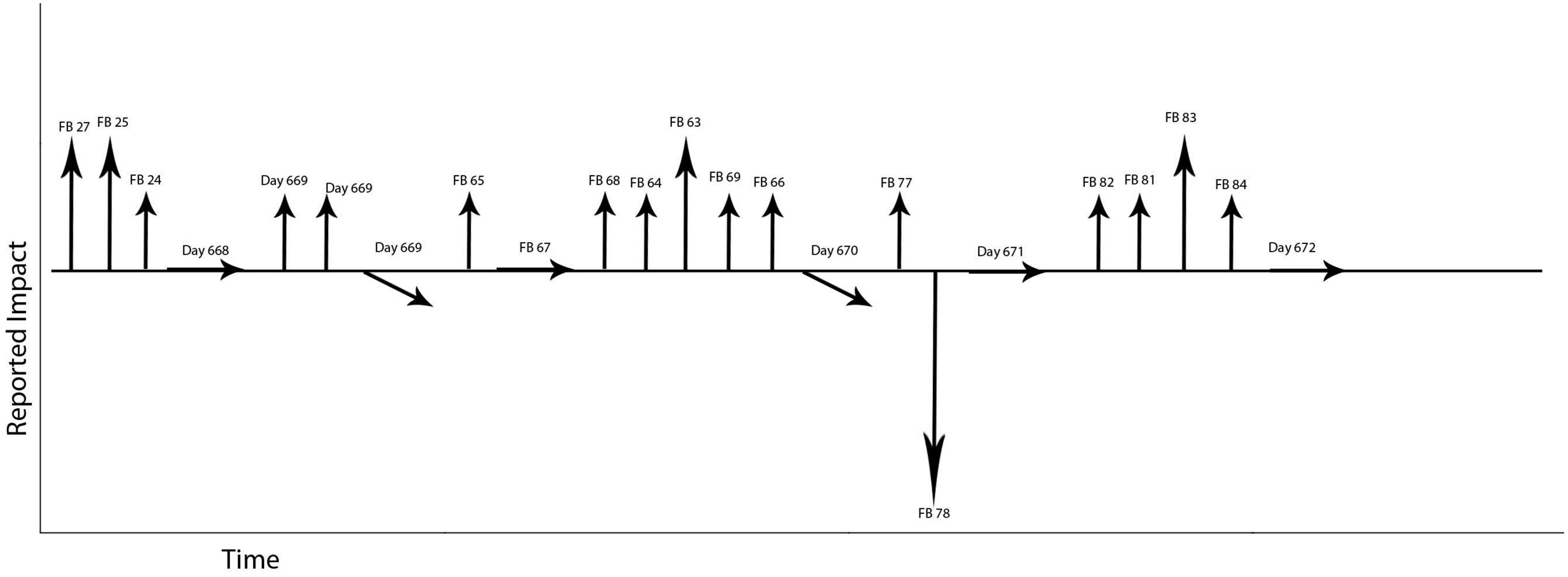
WEEK PATH ANALYSIS FOR PARTICIPANT 15



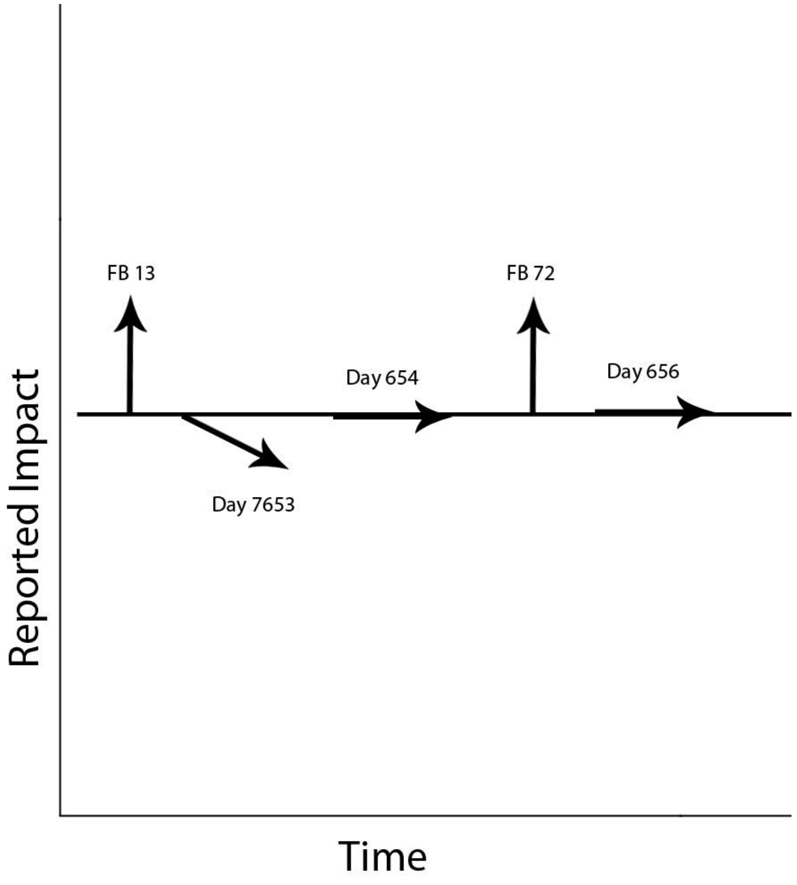
WEEK PATH ANALYSIS FOR PARTICIPANT 16



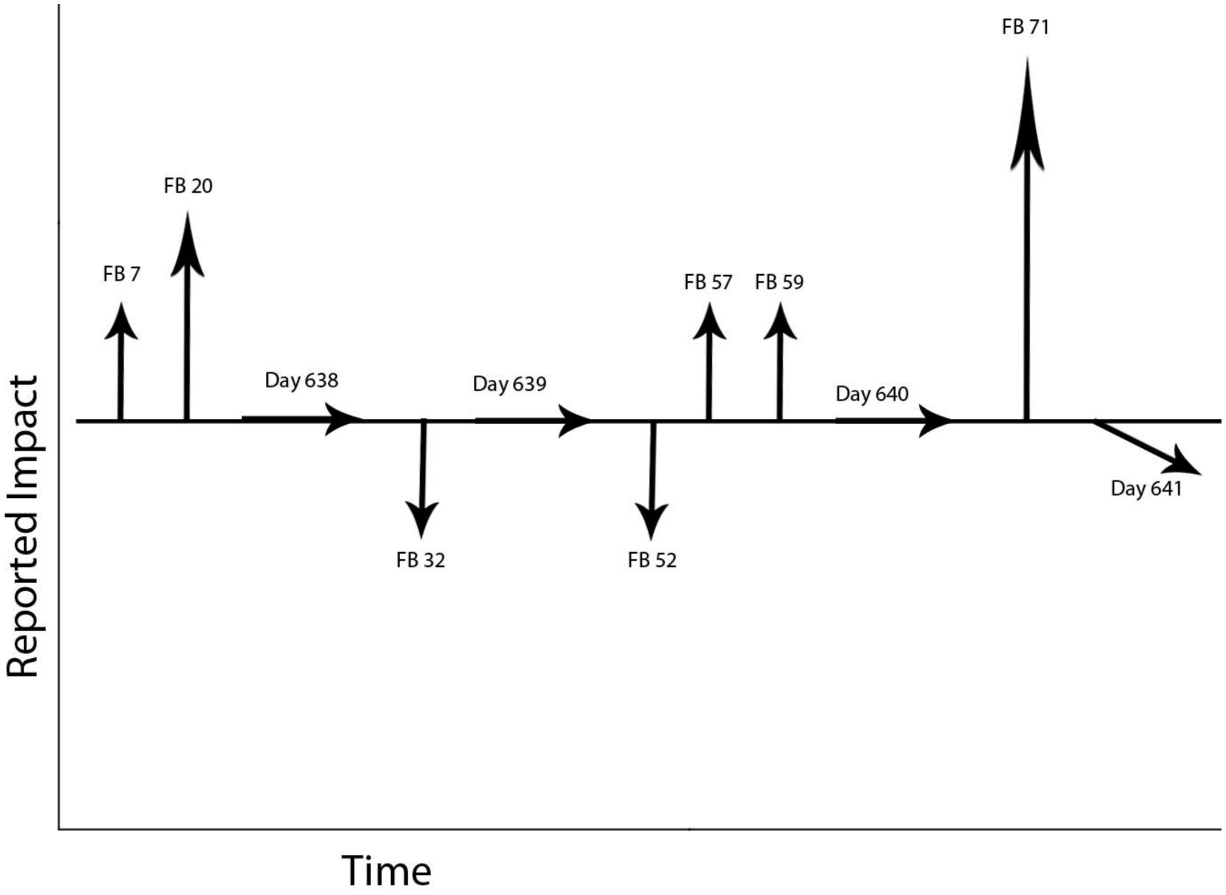
WEEK PATH ANALYSIS FOR PARTICIPANT 19



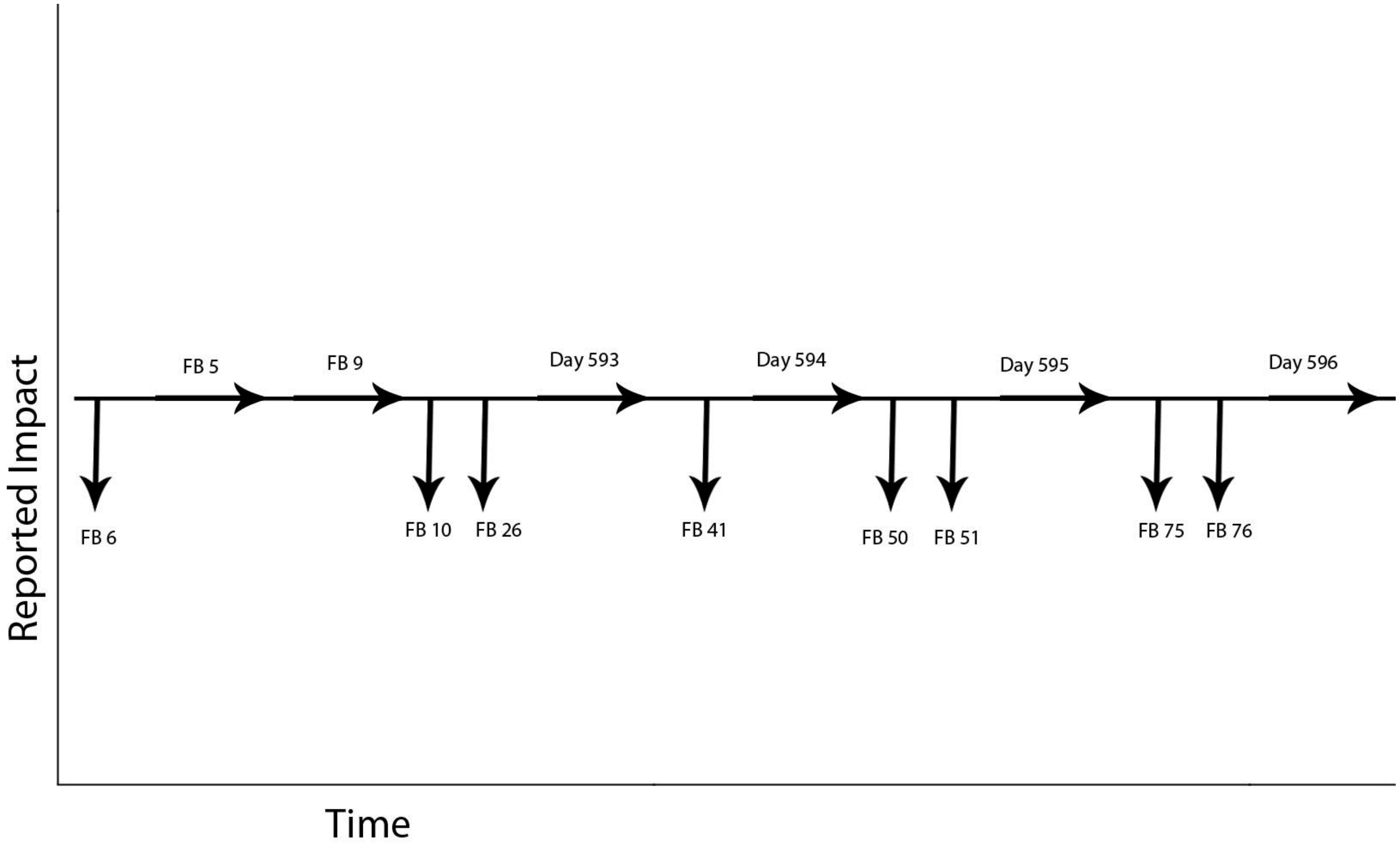
WEEK PATH ANALYSIS FOR PARTICIPANT 20



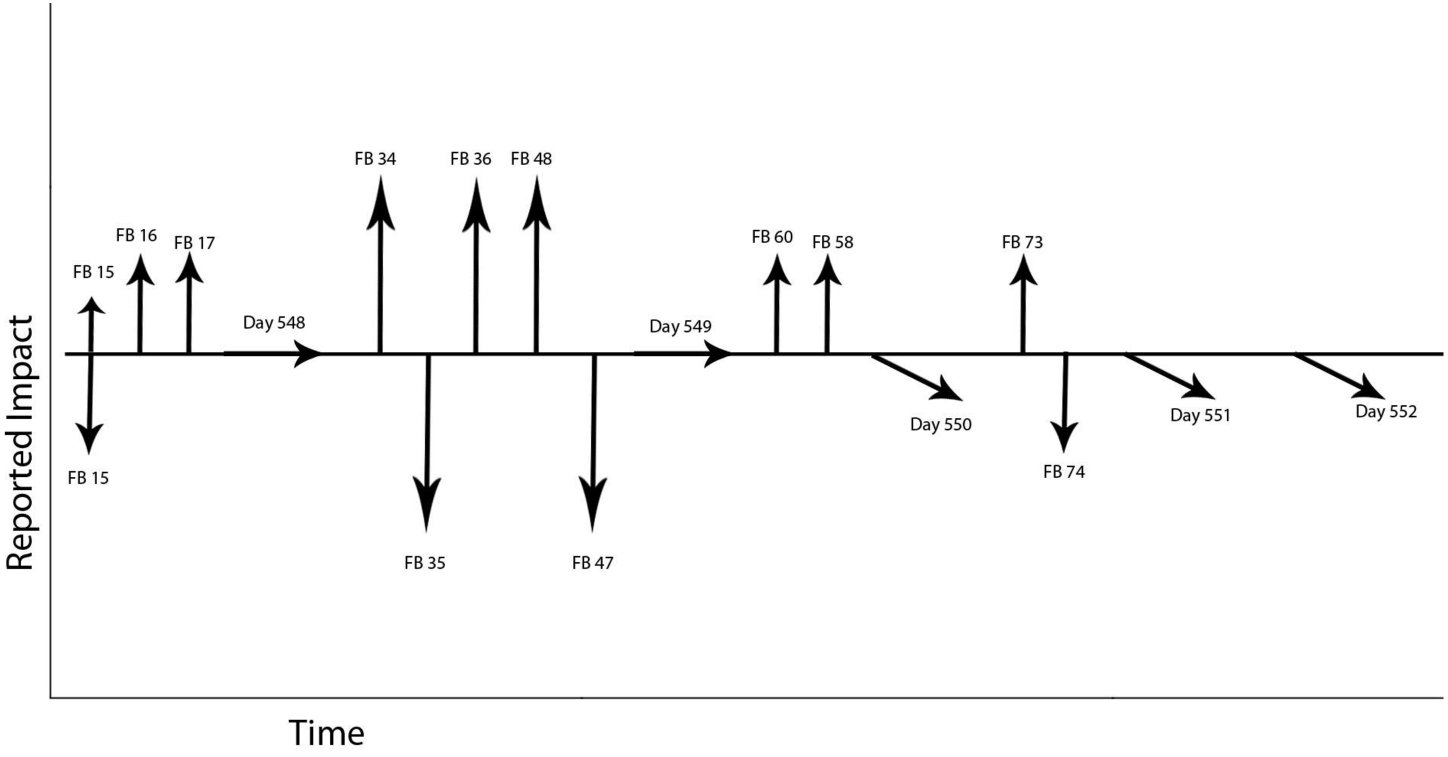
WEEK PATH ANALYSIS FOR PARTICIPANT 21



WEEK PATH ANALYSIS FOR PARTICIPANT 22

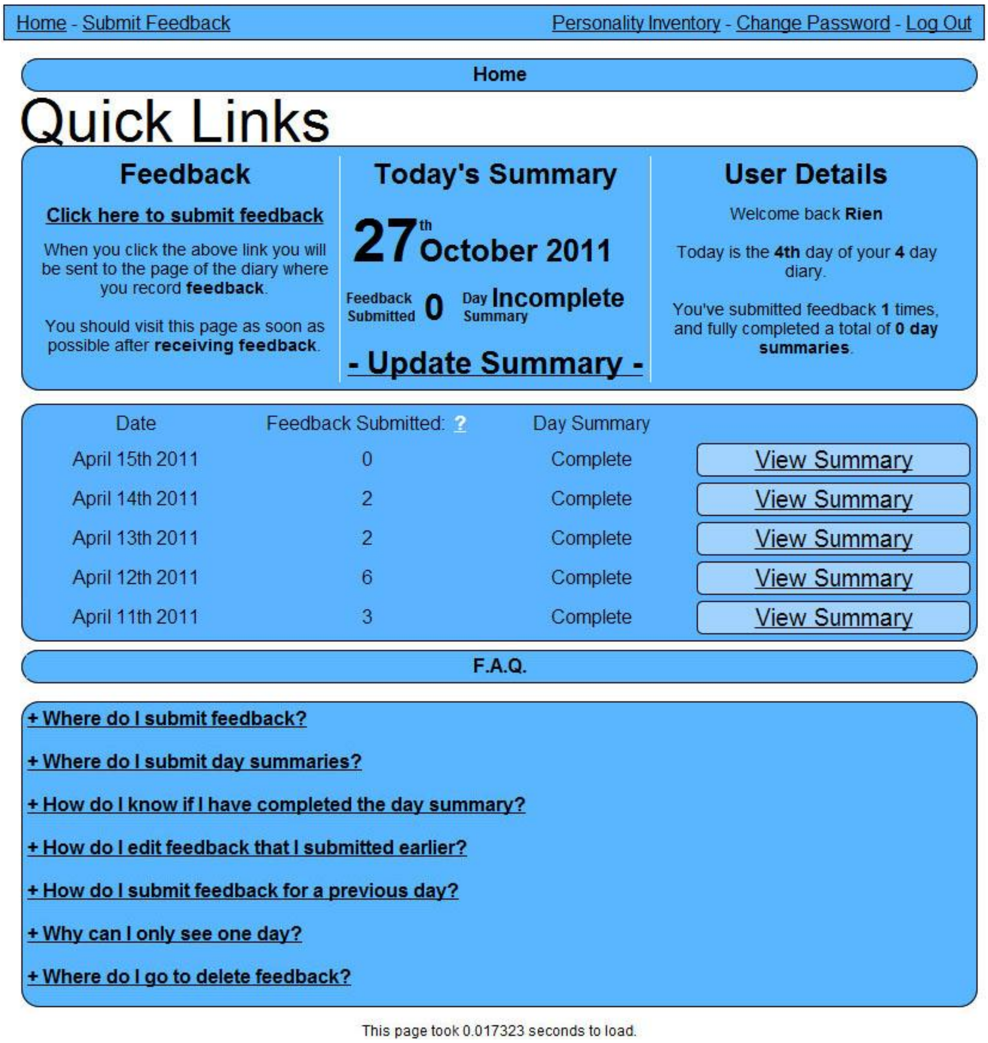


WEEK PATH ANALYSIS FOR PARTICIPANT 25



WEEK PATH ANALYSIS FOR PARTICIPANT 28

APPENDIX B SECTION 7 – SCREENSHOTS OF DIARY STUDY WEB INTERFACE



Quick Links

Feedback

Click here to submit feedback

When you click the above link you will be sent to the page of the diary where you record **feedback**.

You should visit this page as soon as possible after **receiving feedback**.

Today's Summary

27th October 2011

Feedback Submitted 0 Day Summary Incomplete

- Update Summary -

User Details

Welcome back **Rien**

Today is the **4th** day of your **4** day diary.

You've submitted feedback **1** times, and fully completed a total of **0 day summaries**.

Date	Feedback Submitted: ?	Day Summary	
April 15th 2011	0	Complete	View Summary
April 14th 2011	2	Complete	View Summary
April 13th 2011	2	Complete	View Summary
April 12th 2011	6	Complete	View Summary
April 11th 2011	3	Complete	View Summary

+ Where do I submit feedback?

+ Where do I submit day summaries?

+ How do I know if I have completed the day summary?

+ How do I edit feedback that I submitted earlier?

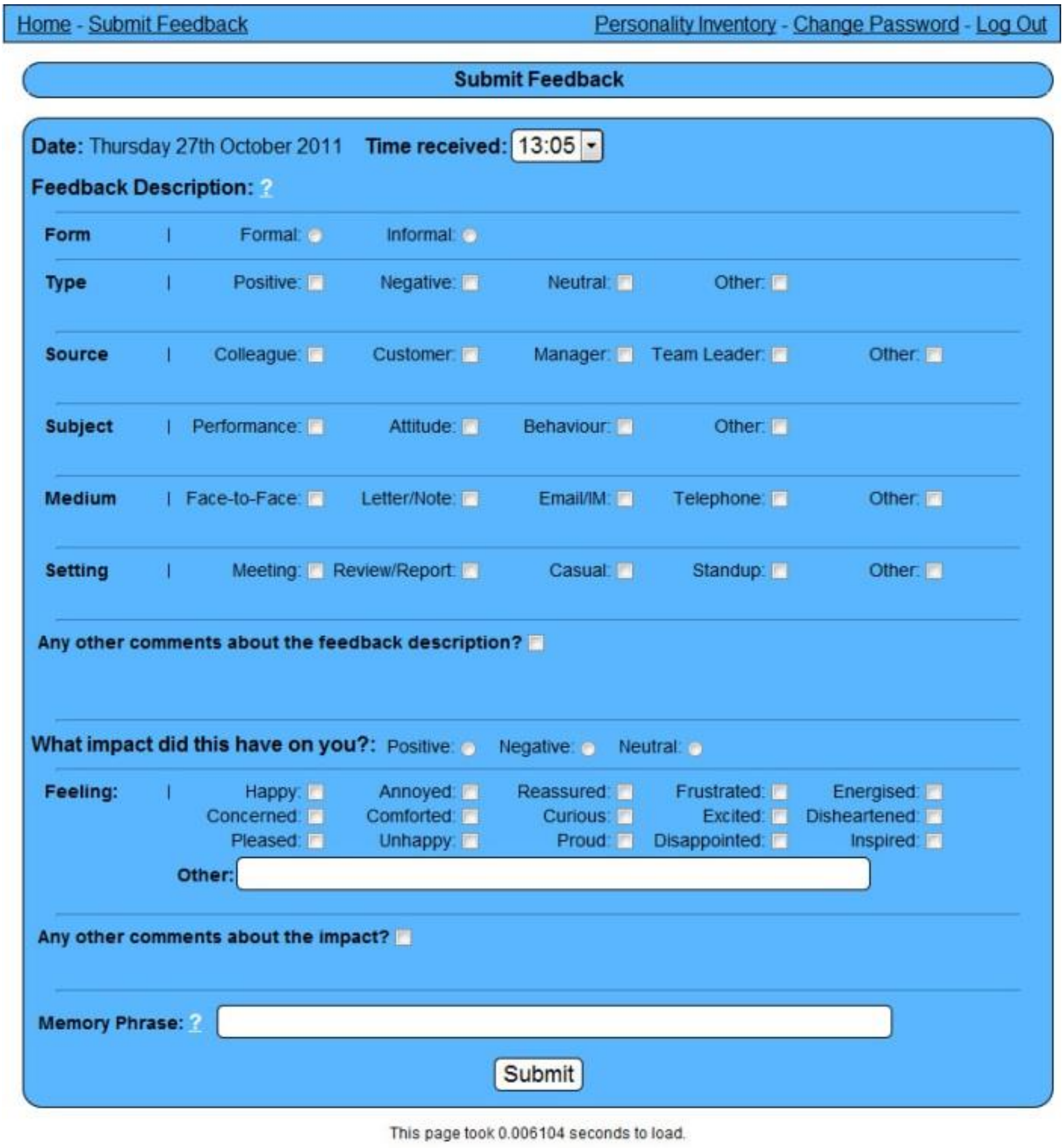
+ How do I submit feedback for a previous day?

+ Why can I only see one day?

+ Where do I go to delete feedback?

This page took 0.017323 seconds to load.

DIARY HOME PAGE



Home - Submit Feedback   Personality Inventory - Change Password - Log Out

Submit Feedback

Date: Thursday 27th October 2011 **Time received:** 13:05

Feedback Description: ?

Form	Formal: ○	Informal: ○			
Type	Positive: □	Negative: □	Neutral: □	Other: □	
Source	Colleague: □	Customer: □	Manager: □	Team Leader: □	Other: □
Subject	Performance: □	Attitude: □	Behaviour: □	Other: □	
Medium	Face-to-Face: □	Letter/Note: □	Email/IM: □	Telephone: □	Other: □
Setting	Meeting: □	Review/Report: □	Casual: □	Standup: □	Other: □

Any other comments about the feedback description? □

What impact did this have on you?: Positive: ○ Negative: ○ Neutral: ○

Feeling:	Happy: □	Annoyed: □	Reassured: □	Frustrated: □	Energised: □
	Concerned: □	Comforted: □	Curious: □	Excited: □	Disheartened: □
	Pleased: □	Unhappy: □	Proud: □	Disappointed: □	Inspired: □

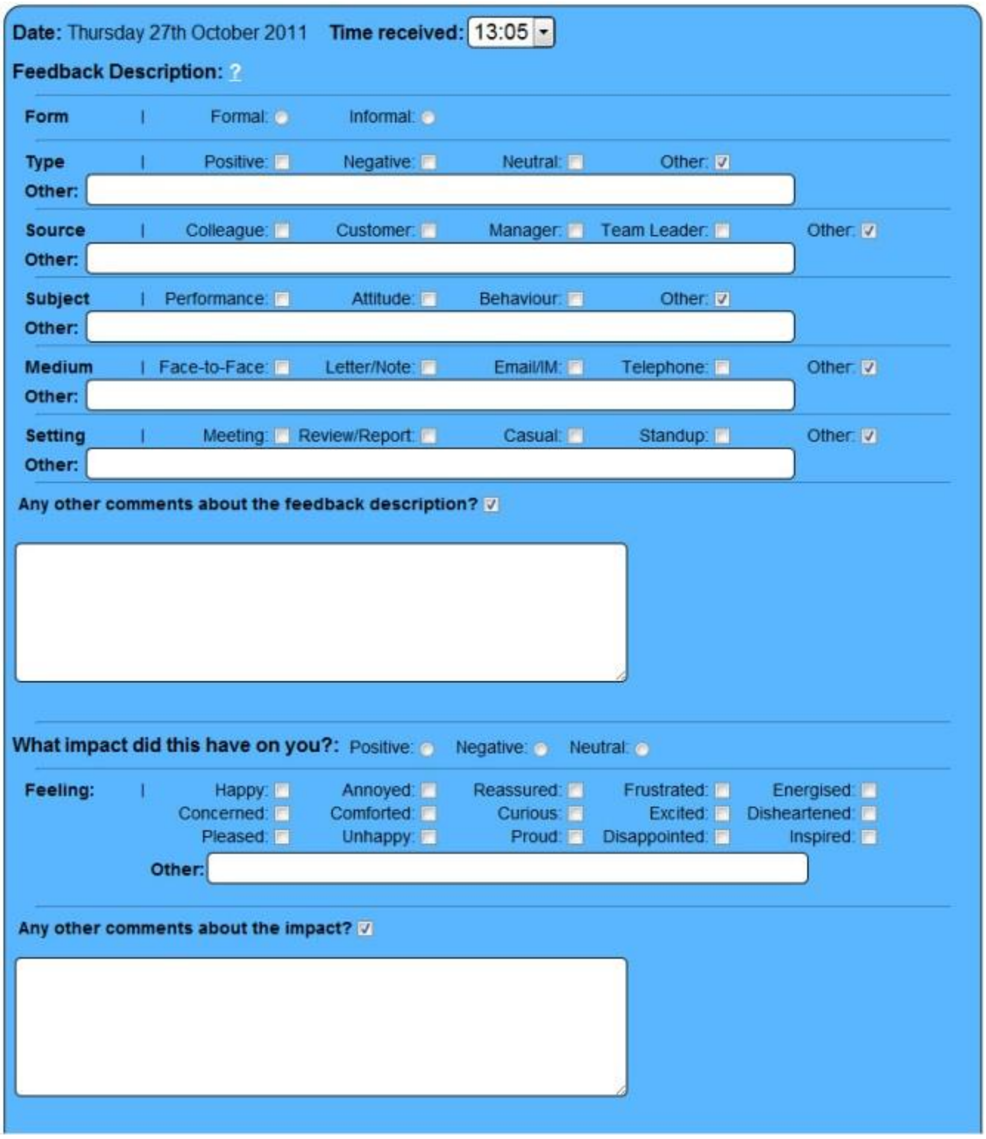
Other:

Any other comments about the impact? □

Memory Phrase: ?

This page took 0.006104 seconds to load.

DIARY FEEDBACK INSTANCE SUBMISSION PAGE



Date: Thursday 27th October 2011 **Time received:** 13:05 ▾

Feedback Description: ?

Form | Formal: ○ Informal: ○

Type | Positive: □ Negative: □ Neutral: □ Other: ☑

Other:

Source | Colleague: □ Customer: □ Manager: □ Team Leader: □ Other: ☑

Other:

Subject | Performance: □ Attitude: □ Behaviour: □ Other: ☑

Other:

Medium | Face-to-Face: □ Letter/Note: □ Email/IM: □ Telephone: □ Other: ☑

Other:

Setting | Meeting: □ Review/Report: □ Casual: □ Standup: □ Other: ☑

Other:

Any other comments about the feedback description? ☑

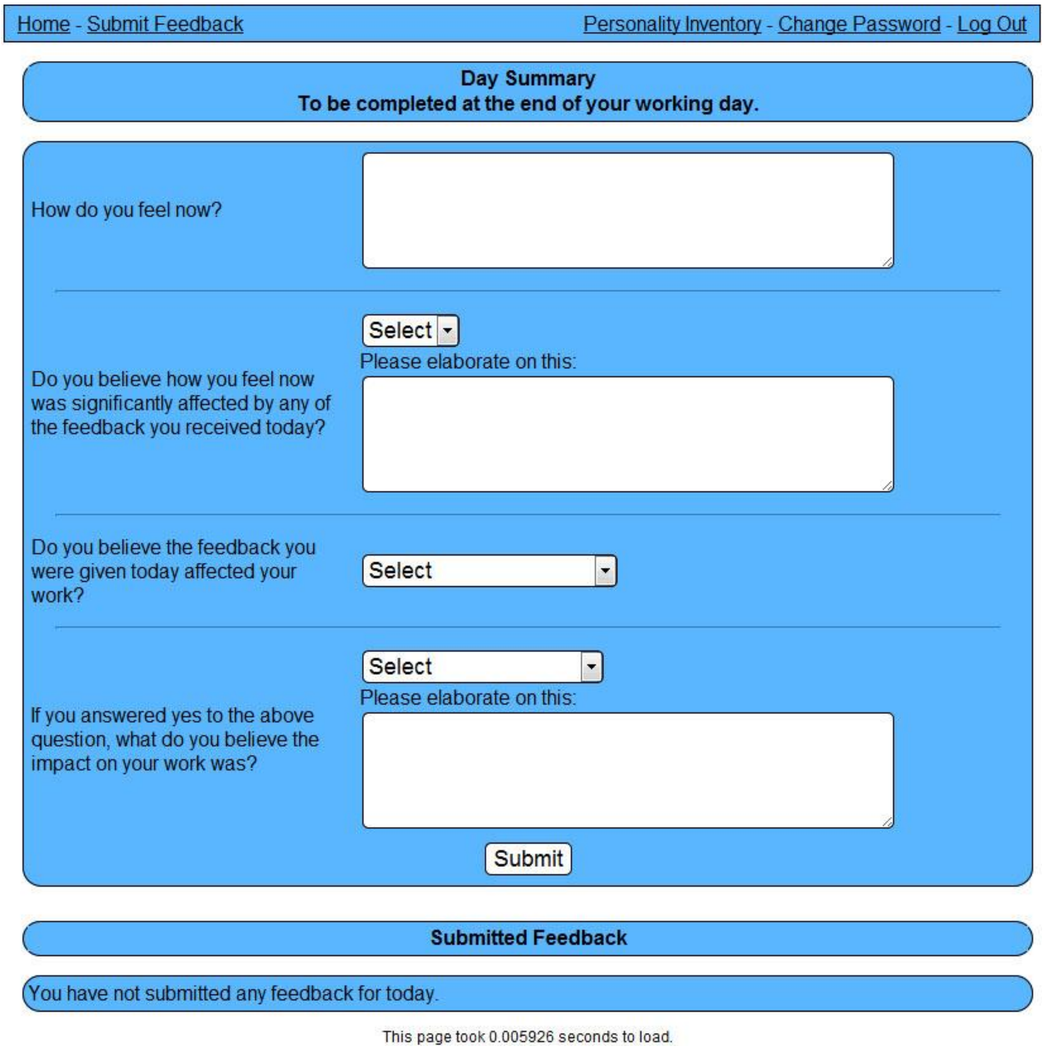
What impact did this have on you?: Positive: ○ Negative: ○ Neutral: ○

Feeling: | Happy: □ Annoyed: □ Reassured: □ Frustrated: □ Energised: □
Concerned: □ Comforted: □ Curious: □ Excited: □ Disheartened: □
Pleased: □ Unhappy: □ Proud: □ Disappointed: □ Inspired: □

Other:

Any other comments about the impact? ☑

DIARY FEEDBACK INSTANCE SUBMISSION PAGE WITH ADDITIONAL BOXES



Day Summary
To be completed at the end of your working day.

How do you feel now?

Do you believe how you feel now was significantly affected by any of the feedback you received today?

Please elaborate on this:

Do you believe the feedback you were given today affected your work?

If you answered yes to the above question, what do you believe the impact on your work was?

Please elaborate on this:

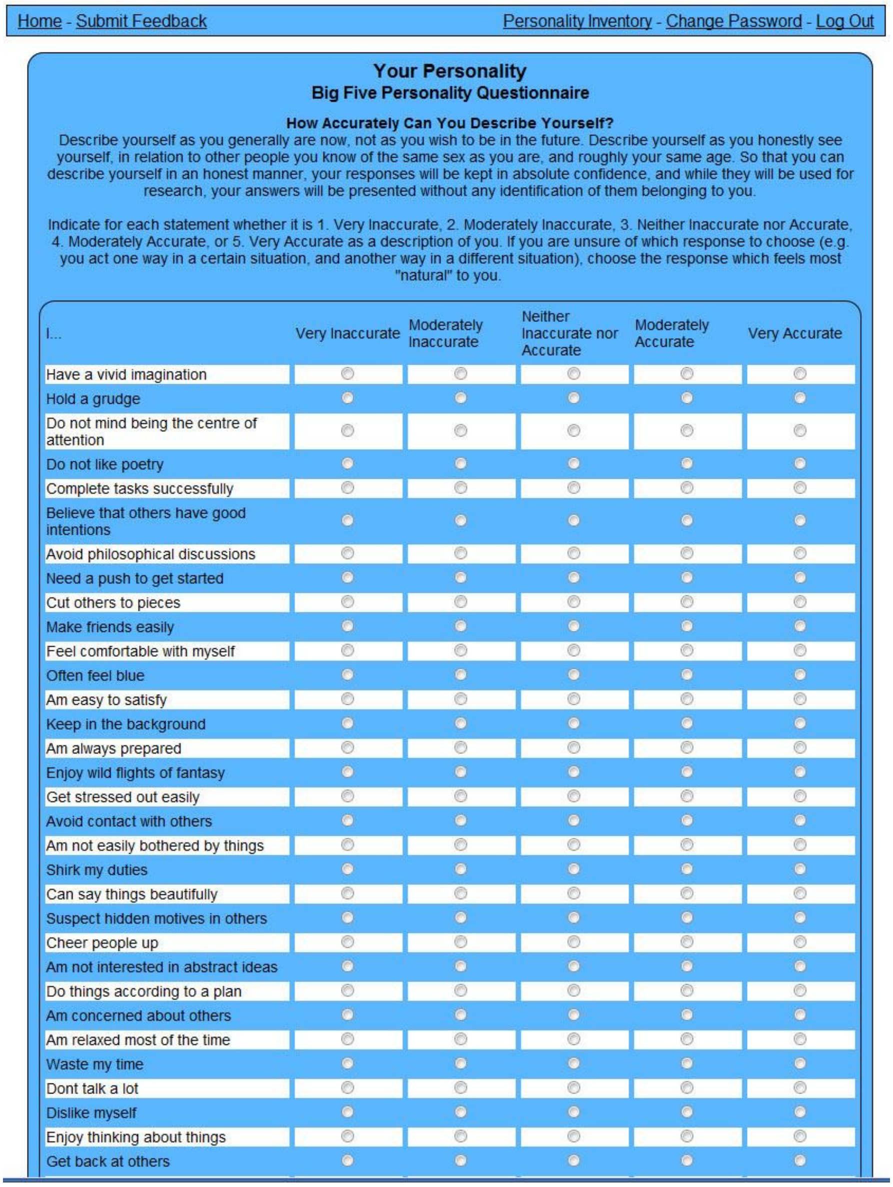
Submitted Feedback

You have not submitted any feedback for today.

This page took 0.005926 seconds to load.

DIARY END OF DAY SUMMARY PAGE

APPENDIX B SECTION 8 – SCREENSHOT OF PERSONALITY INVENTORY INTERFACE



Your Personality

Big Five Personality Questionnaire

How Accurately Can You Describe Yourself?

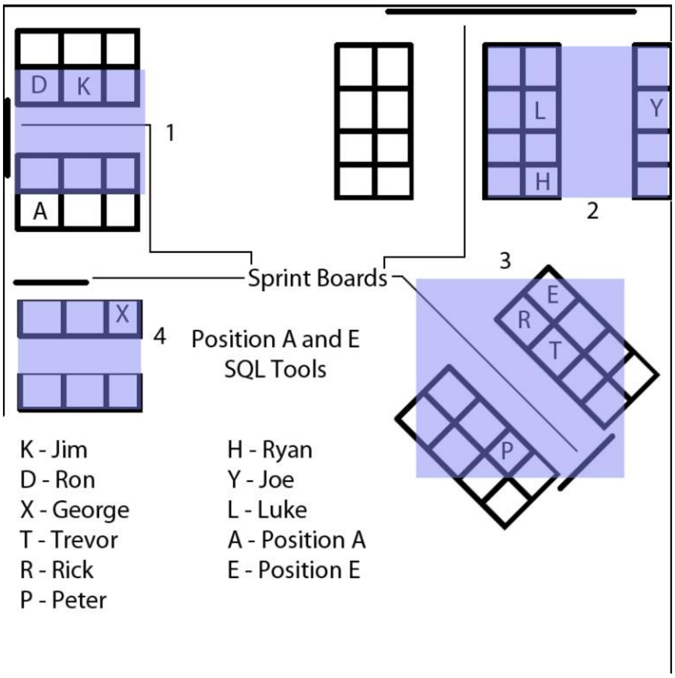
Describe yourself as you generally are now, not as you wish to be in the future. Describe yourself as you honestly see yourself, in relation to other people you know of the same sex as you are, and roughly your same age. So that you can describe yourself in an honest manner, your responses will be kept in absolute confidence, and while they will be used for research, your answers will be presented without any identification of them belonging to you.

Indicate for each statement whether it is 1. Very Inaccurate, 2. Moderately Inaccurate, 3. Neither Inaccurate nor Accurate, 4. Moderately Accurate, or 5. Very Accurate as a description of you. If you are unsure of which response to choose (e.g. you act one way in a certain situation, and another way in a different situation), choose the response which feels most "natural" to you.

I...	Very Inaccurate	Moderately Inaccurate	Neither Inaccurate nor Accurate	Moderately Accurate	Very Accurate
Have a vivid imagination	○	○	○	○	○
Hold a grudge	○	○	○	○	○
Do not mind being the centre of attention	○	○	○	○	○
Do not like poetry	○	○	○	○	○
Complete tasks successfully	○	○	○	○	○
Believe that others have good intentions	○	○	○	○	○
Avoid philosophical discussions	○	○	○	○	○
Need a push to get started	○	○	○	○	○
Cut others to pieces	○	○	○	○	○
Make friends easily	○	○	○	○	○
Feel comfortable with myself	○	○	○	○	○
Often feel blue	○	○	○	○	○
Am easy to satisfy	○	○	○	○	○
Keep in the background	○	○	○	○	○
Am always prepared	○	○	○	○	○
Enjoy wild flights of fantasy	○	○	○	○	○
Get stressed out easily	○	○	○	○	○
Avoid contact with others	○	○	○	○	○
Am not easily bothered by things	○	○	○	○	○
Shirk my duties	○	○	○	○	○
Can say things beautifully	○	○	○	○	○
Suspect hidden motives in others	○	○	○	○	○
Cheer people up	○	○	○	○	○
Am not interested in abstract ideas	○	○	○	○	○
Do things according to a plan	○	○	○	○	○
Am concerned about others	○	○	○	○	○
Am relaxed most of the time	○	○	○	○	○
Waste my time	○	○	○	○	○
Dont talk a lot	○	○	○	○	○
Dislike myself	○	○	○	○	○
Enjoy thinking about things	○	○	○	○	○
Get back at others	○	○	○	○	○

PERSONALITY INVENTORY INSTRUMENT

APPENDIX B SECTION 9 – IMAGES AND PHOTOS OF RED GATE ENVIRONMENT



SQL TEAM PARTICIPANTS AND RESEARCH POSITION LOCATIONS

EXAMPLE RED GATE OFFICE ENVIRONMENT

RED GATE SOFTWARE ENGINEERING ALTERNATIVE CHAIR

RED GATE ENVIRONMENT IMAGES

THREE RED GATE ENVIRONMENT IMAGES

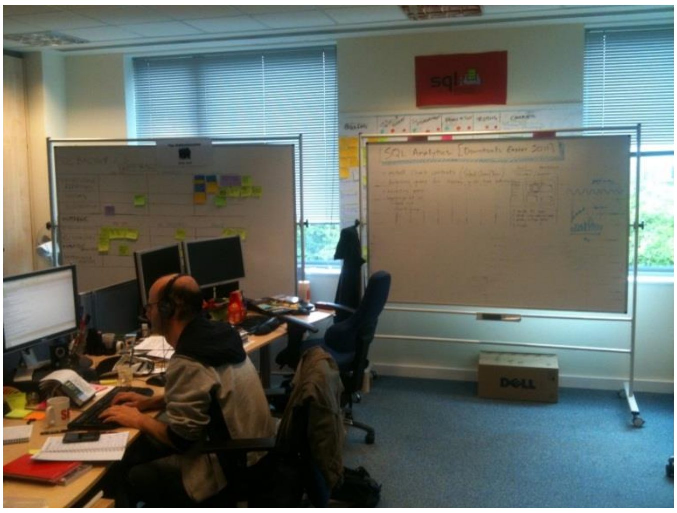
RED GATE SOFTWARE DEVELOPMENT ENVIRONMENT

TEAM 1 WORKING ENVIRONMENT

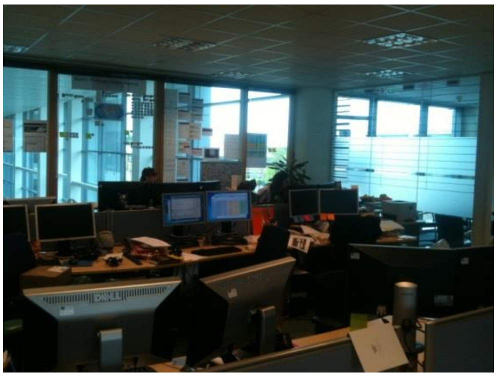
TEAM 2 WORKING ENVIRONMENT

TEAM 3 WORKING ENVIRONMENT

TEAM 4 WORKING ENVIRONMENT

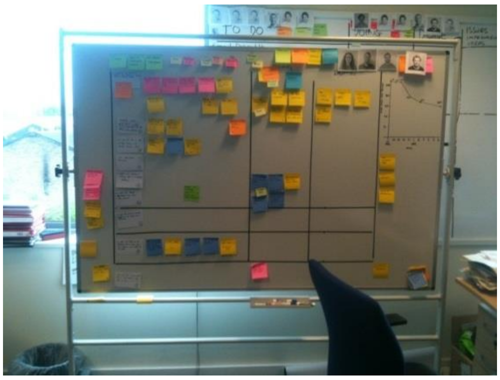


DATABASE TEAM SPRINT BOARDS



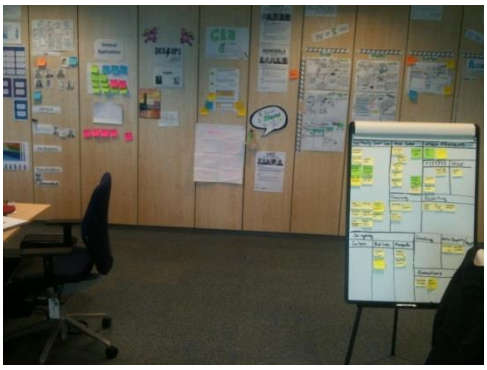
NEW BUSINESS SPRINT BOARD AND WINDOW DISPLAY OF INFORMATION

NERF GUN



DEVOPS SPRINT BOARD

– DEVOPS TEAM ENVIRONMENT



DEVOPS WALL OF INFORMATION

DEVOPS WALL OF INFORMATION

FEEDBACK ABOVE SPRINT BOARD

Epic!
You're delivering quality tools that provide a ton of v to us "in the tren developers.
To Do

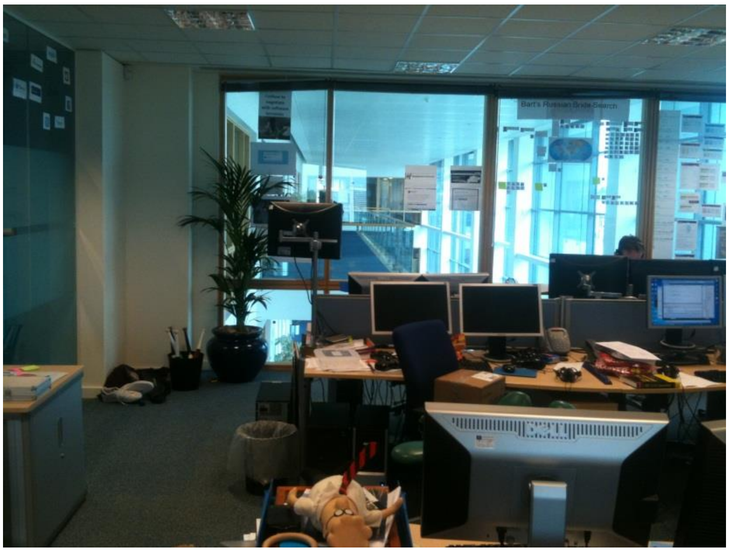
FEEDBACK ABOVE SPRINT BOARD

FEEDBACK BEHIND DESKS

DATABASE DIVISION ENVIRONMENT

This was perhaps the quickest I have ever gotten a response from a software support specialist, very good work. I would only add that I am only a little surprised because Redgate Tools have impressed me with their features, ability and quality, and I fully expected your customer support to be as excellent.
Hey, FYI, the patch works brilliantly as well. Thank you! Tell your boss Red Gate has the best support on the planet, I love you guys!

PILLAR OF FEEDBACK



NEW BUSINESS TEAM ENVIRONMENT

DEVOPS WALL

APPENDIX C SECTION 1 – THE SHORT-TERM IMPACT OF RECEIVED FEEDBACK SCENARIO FORM

FEEDBACK IMPACT

Introduction

As part of this research we're trying to understand more about the impact feedback can have on software engineers. Listed below are some fictitious people, but their role description should be similar to someone you work with. Please read through all of the descriptions, and then read through each of the questions and think about how each scenario would impact you. After you've read each scenario please tick any of the boxes that you believe represent the impact of this scenario.

Please bring a print-out of the completed form with you when we meet.

People

Tom – Project Manager. Tom sits in the same area as you.
Rick – Software Engineer. Rick is a colleague of yours in your team.
Simon – Software Tester. Simon is a colleague of yours in your team.
Gary – Software Engineer. Gary is a colleague of yours in another team.
Boris – Division Head. Boris doesn't sit with your team or any team in your division.
Craig – CEO. Craig has a desk he sometimes uses in your division.

Questions

1. You're working with Rick (Software Engineer) on a piece of code he's having trouble with. After helping him, he thanks you and tells you what a life saver you are. Would this scenario have an impact on your:

Attitude □
Behaviour □
Motivation □
Productivity □
Job Satisfaction □

2. At a stand up meeting, Tom (Project Manager) tells the team how he's disappointed with the current progress on part of the project. You feel responsible for this lack of progress. Would this scenario have an impact on your:

Attitude □
Behaviour □
Motivation □
Productivity □
Job Satisfaction □

3. You receive an email from Boris (Division Head) telling you what a brilliant job you've done lately and how he's impressed with your performance. Would this scenario have an impact on your:

Attitude □
Behaviour □
Motivation □
Productivity □
Job Satisfaction □

4. During a one to one meeting with Tom (Project Manager), he talks to you about a problem with your work on a recent project. You suspect Tom (Project Manager) is relaying feedback from Rick (Software Engineer). Would this scenario have an impact on your:

Attitude □
Behaviour □
Motivation □
Productivity □
Job Satisfaction □

5. Gary (Software Engineer) asks you to help him with a problem he's stuck on. After you help him he thanks you, and you overhear him telling Simon (Software Tester) what a great help you've been. Would this scenario have an impact on your:

Attitude □
Behaviour □
Motivation □
Productivity □
Job Satisfaction □

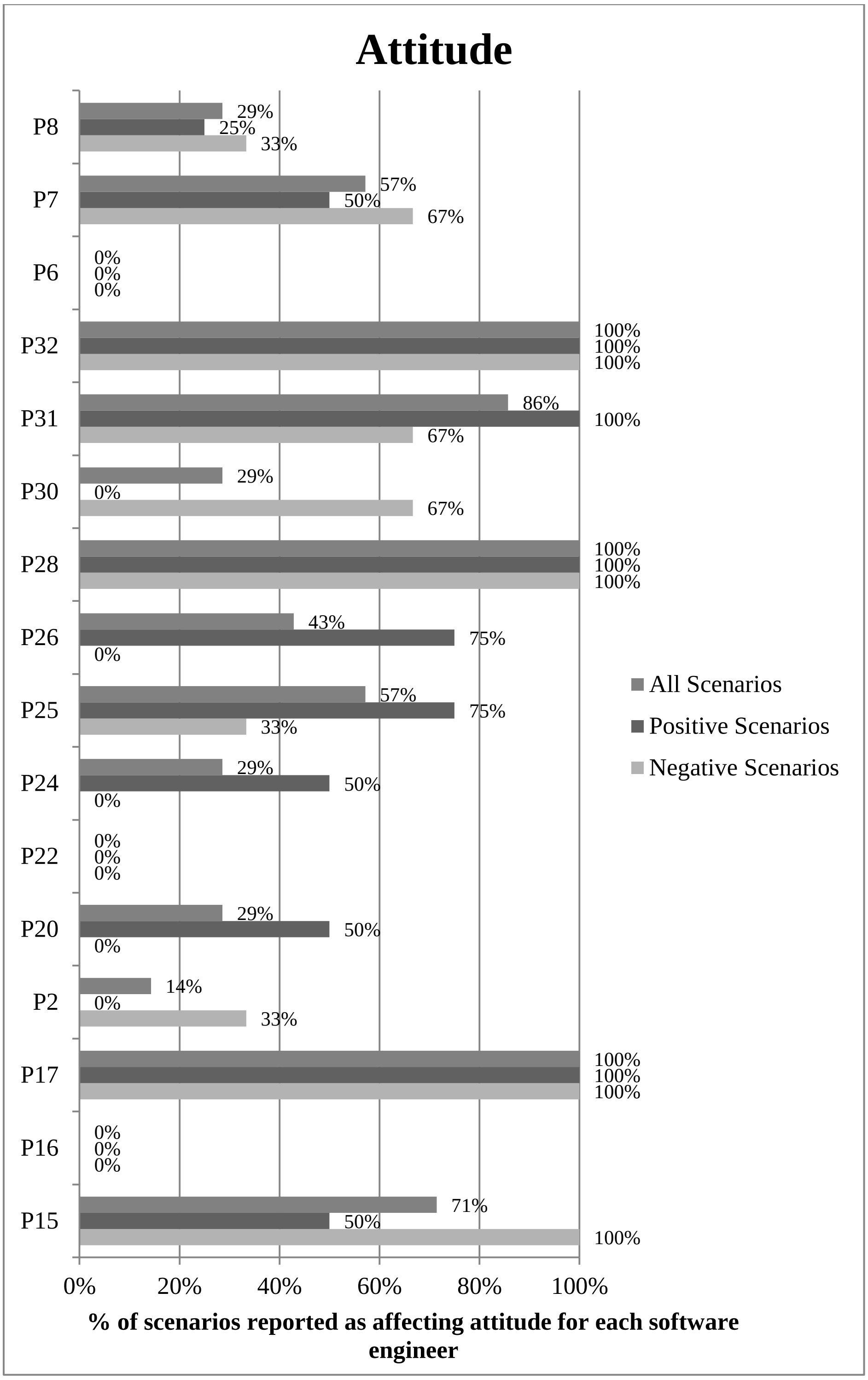
6. Simon (Software Tester) comes over to speak to you. He has some bad news – recent changes you made broke the system. Would this scenario have an impact on your:

Attitude □
Behaviour □
Motivation □
Productivity □
Job Satisfaction □

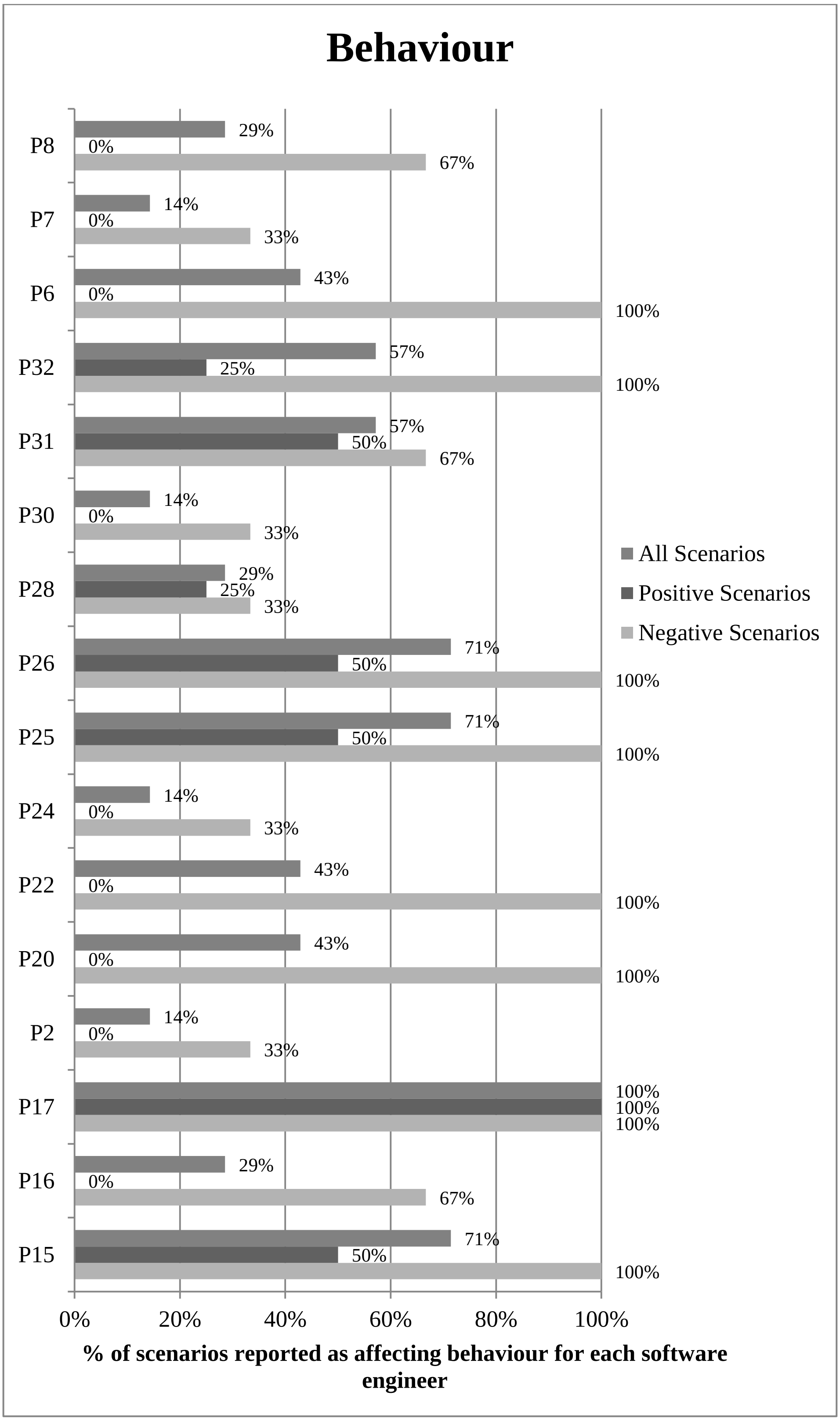
7. During a one to one meeting with Tom (Project Manager), he tells you how happy he is with your recent performance. Would this scenario have an impact on your:

Attitude □
Behaviour □
Motivation □
Productivity □
Job Satisfaction □

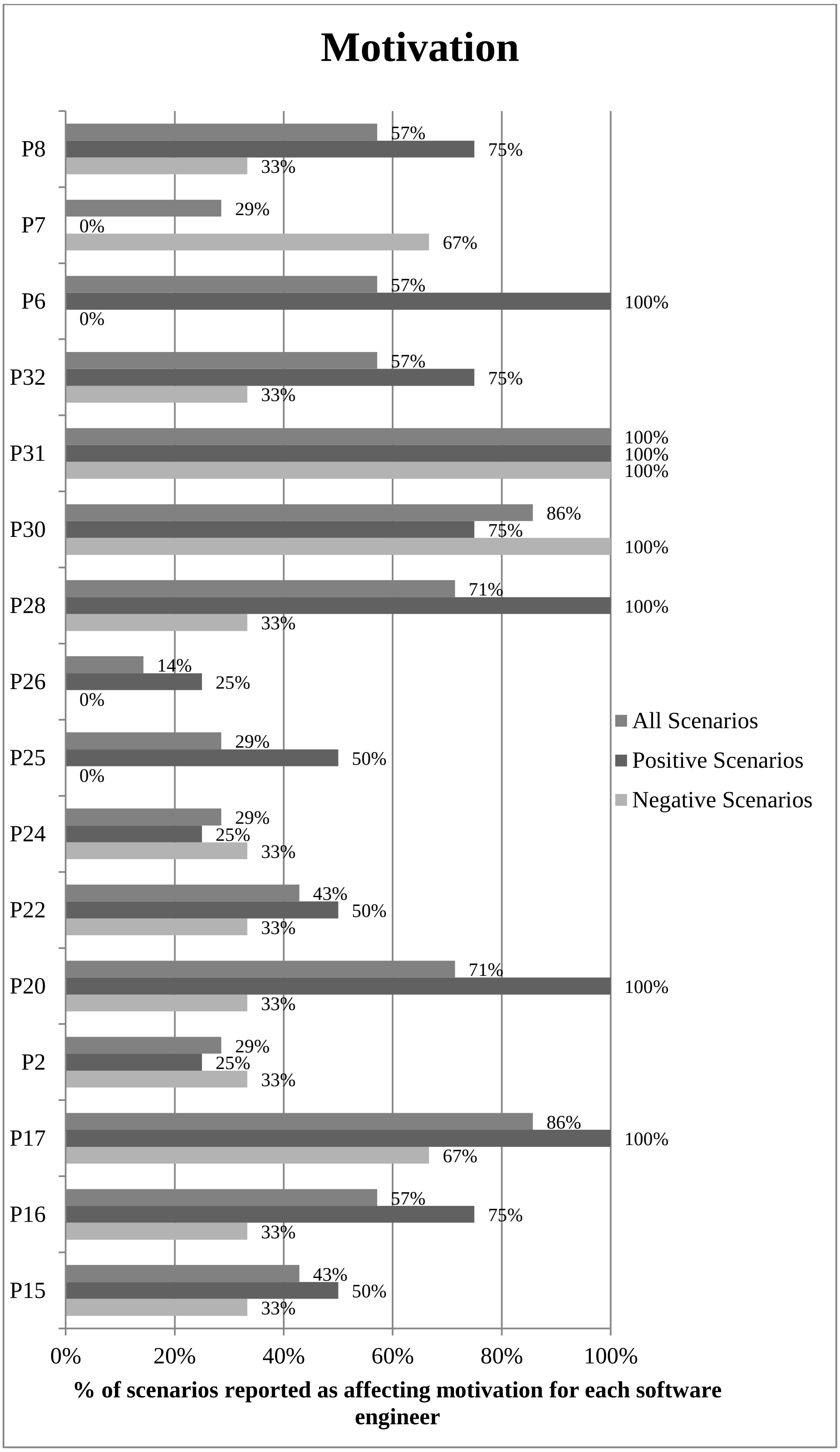
APPENDIX C SECTION 2 – ANALYSIS CHARTS INVESTIGATING INDIVIDUAL RESPONSES TO POSITIVE, NEGATIVE, AND ALL FEEDBACK SCENARIOS



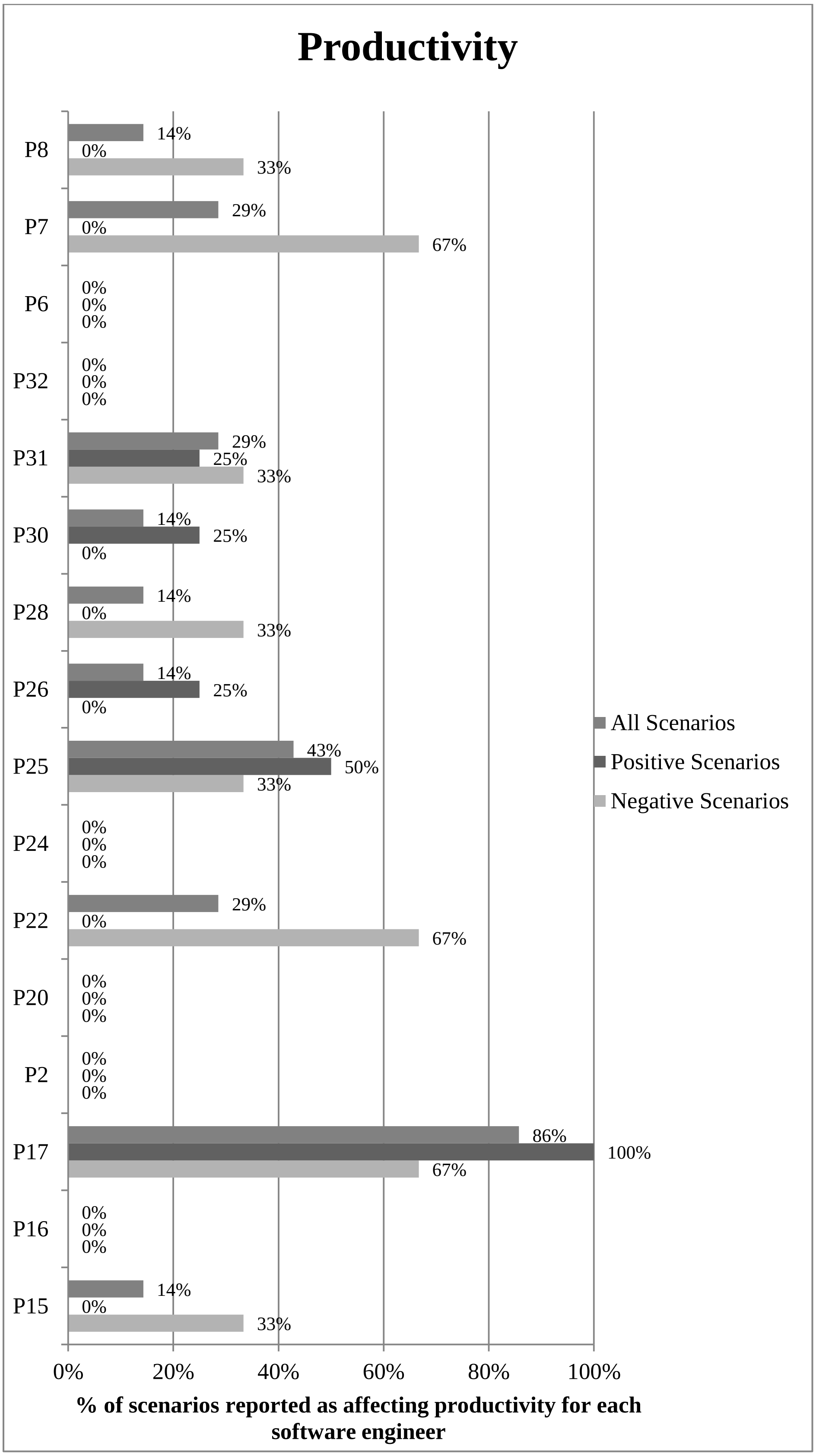
SOFTWARE ENGINEER INDICATION OF AFFECT ON ATTITUDE OF RECEIVED FEEDBACK



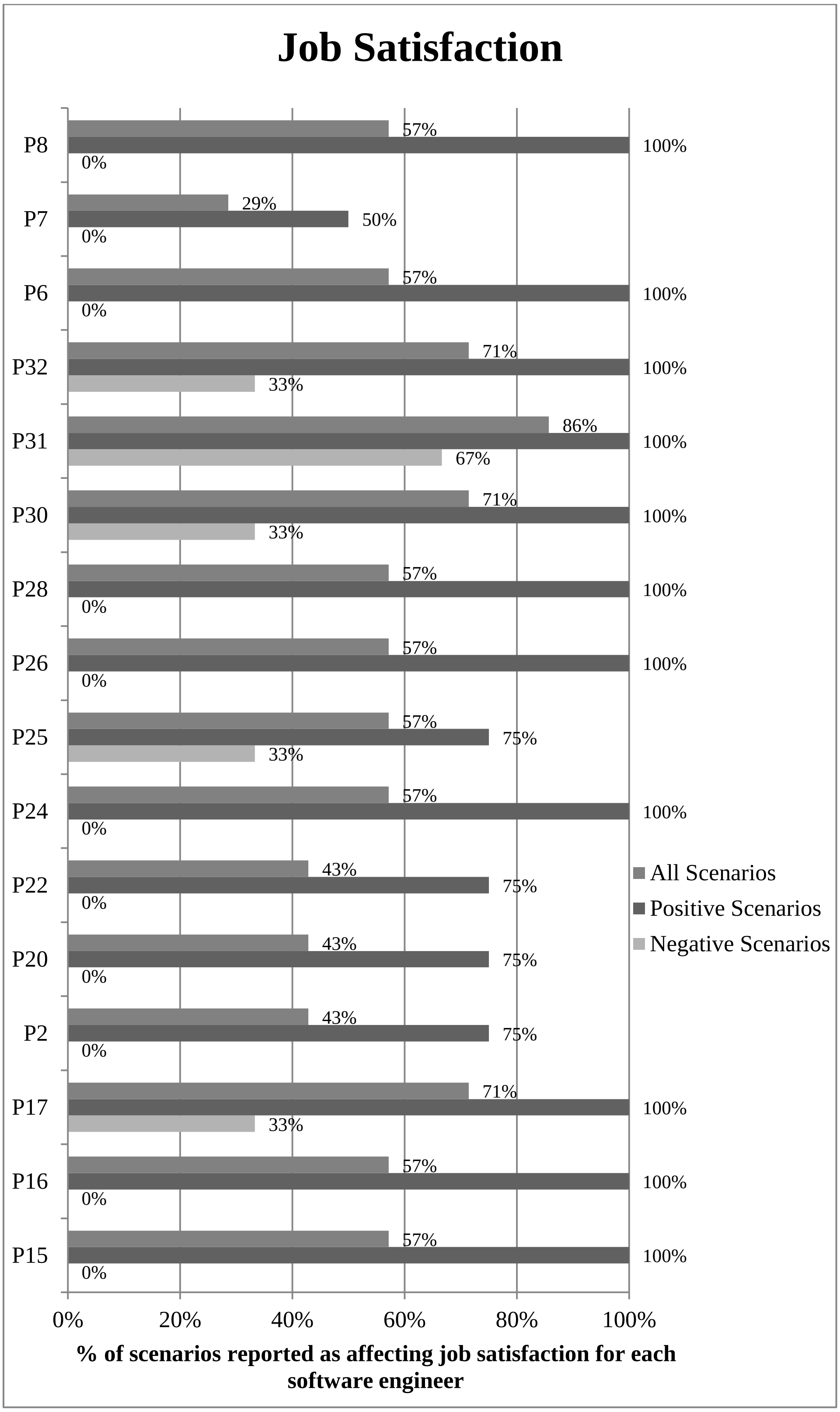
SOFTWARE ENGINEER INDICATION OF AFFECT ON BEHAVIOUR OF RECEIVED FEEDBACK



SOFTWARE ENGINEER INDICATION OF AFFECT ON MOTIVATION OF RECEIVED FEEDBACK

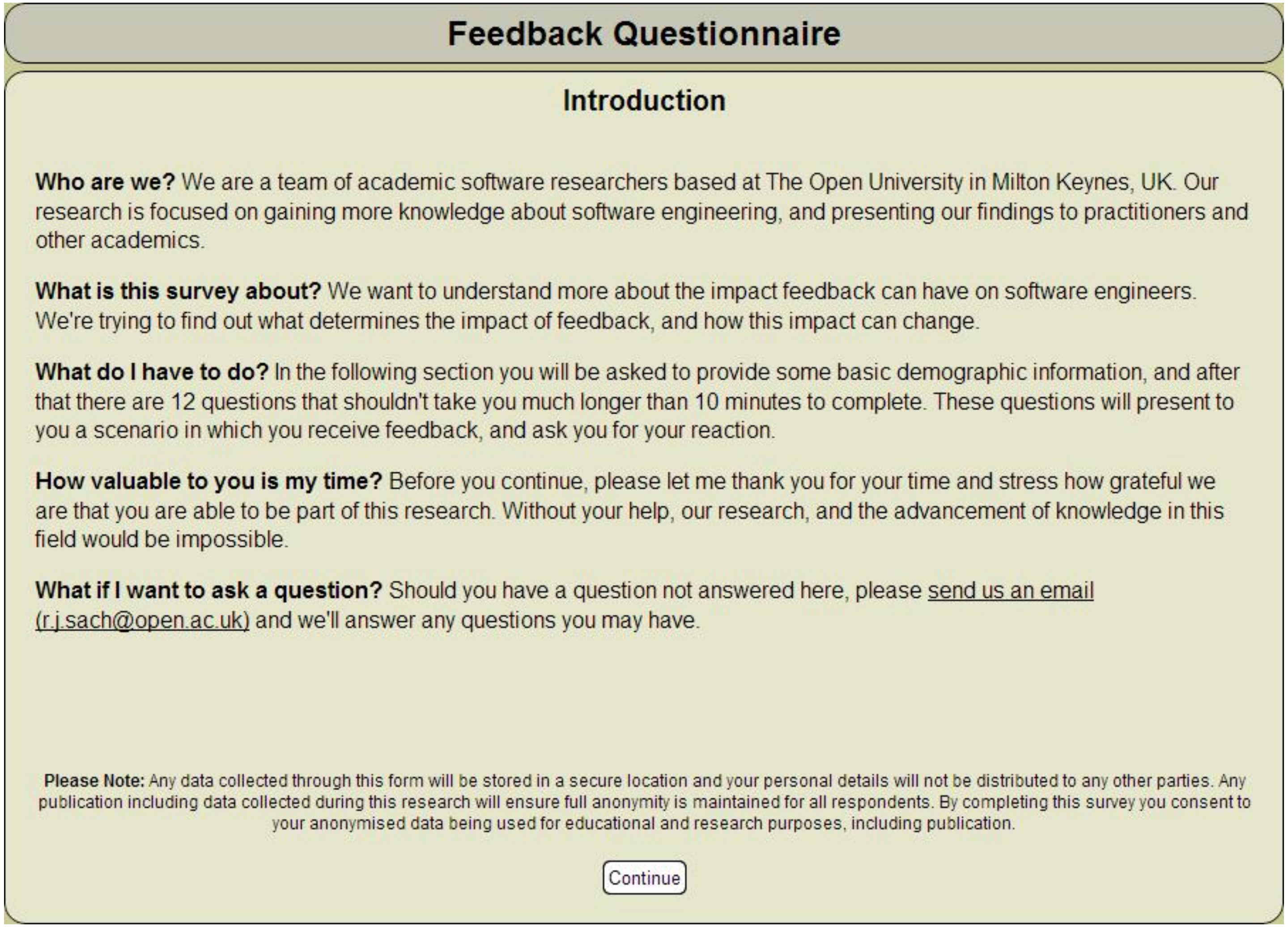


SOFTWARE ENGINEER INDICATION OF AFFECT ON PRODUCTIVITY OF RECEIVED FEEDBACK



SOFTWARE ENGINEER INDICATION OF AFFECT ON JOB SATISFACTION OF RECEIVED FEEDBACK

APPENDIX D SECTION 1 – THE EFFECT OF ‘SOURCE’ AND ‘MEDIUM’ FEEDBACK CHARACTERISTICS ONLINE SURVEY PAGE-BY-PAGE REPLICA



Feedback Questionnaire

Introduction

Who are we? We are a team of academic software researchers based at The Open University in Milton Keynes, UK. Our research is focused on gaining more knowledge about software engineering, and presenting our findings to practitioners and other academics.

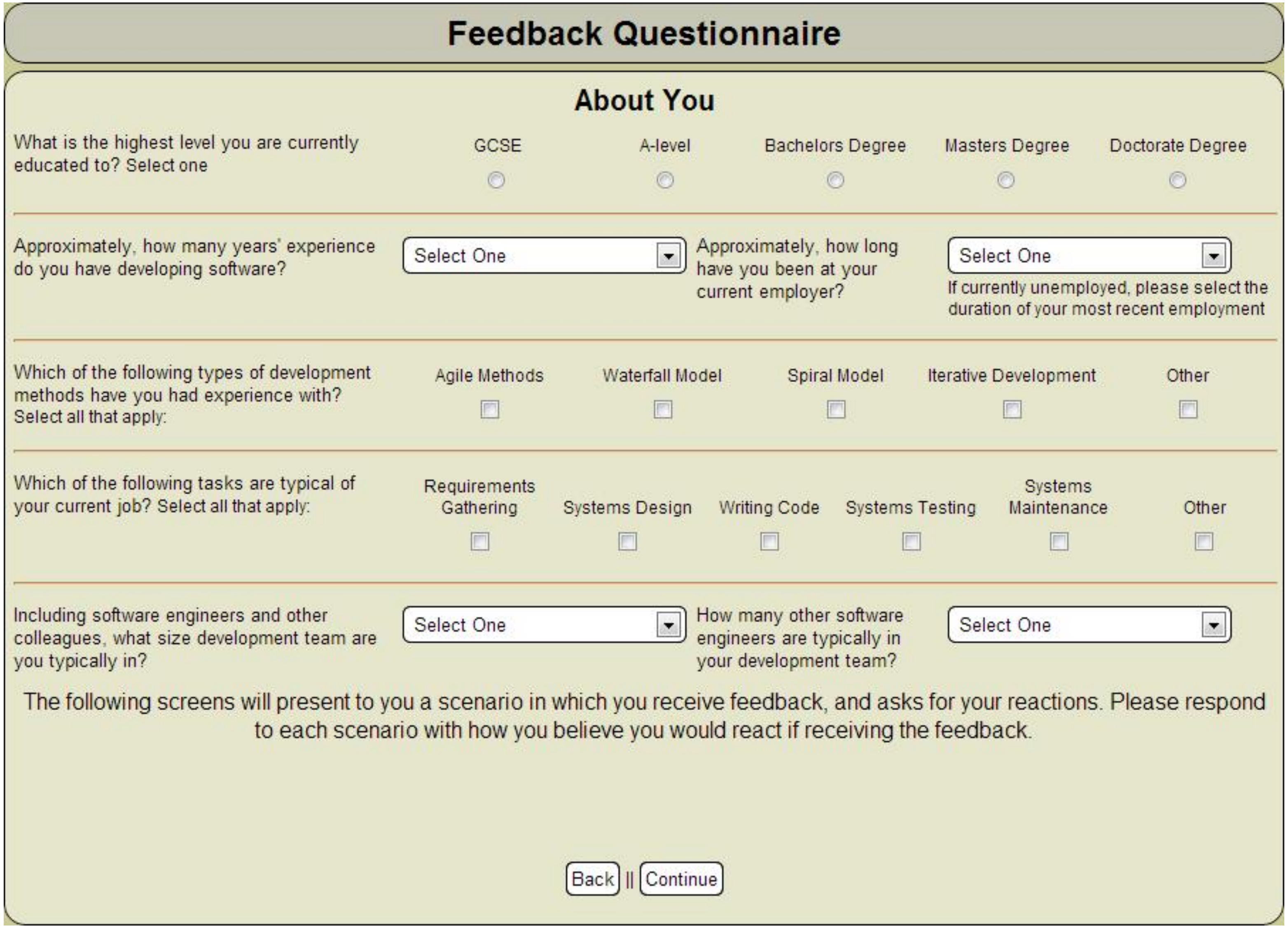
What is this survey about? We want to understand more about the impact feedback can have on software engineers. We're trying to find out what determines the impact of feedback, and how this impact can change.

What do I have to do? In the following section you will be asked to provide some basic demographic information, and after that there are 12 questions that shouldn't take you much longer than 10 minutes to complete. These questions will present to you a scenario in which you receive feedback, and ask you for your reaction.

How valuable to you is my time? Before you continue, please let me thank you for your time and stress how grateful we are that you are able to be part of this research. Without your help, our research, and the advancement of knowledge in this field would be impossible.

What if I want to ask a question? Should you have a question not answered here, please send us an email (r.j.sach@open.ac.uk) and we'll answer any questions you may have.

Please Note: Any data collected through this form will be stored in a secure location and your personal details will not be distributed to any other parties. Any publication including data collected during this research will ensure full anonymity is maintained for all respondents. By completing this survey you consent to your anonymised data being used for educational and research purposes, including publication.



Feedback Questionnaire

About You

What is the highest level you are currently educated to? Select one	GCSE ○	A-level ○	Bachelors Degree ○	Masters Degree ○	Doctorate Degree ○

Approximately, how many years' experience do you have developing software? Select One ▾

Approximately, how long have you been at your current employer? Select One ▾

If currently unemployed, please select the duration of your most recent employment

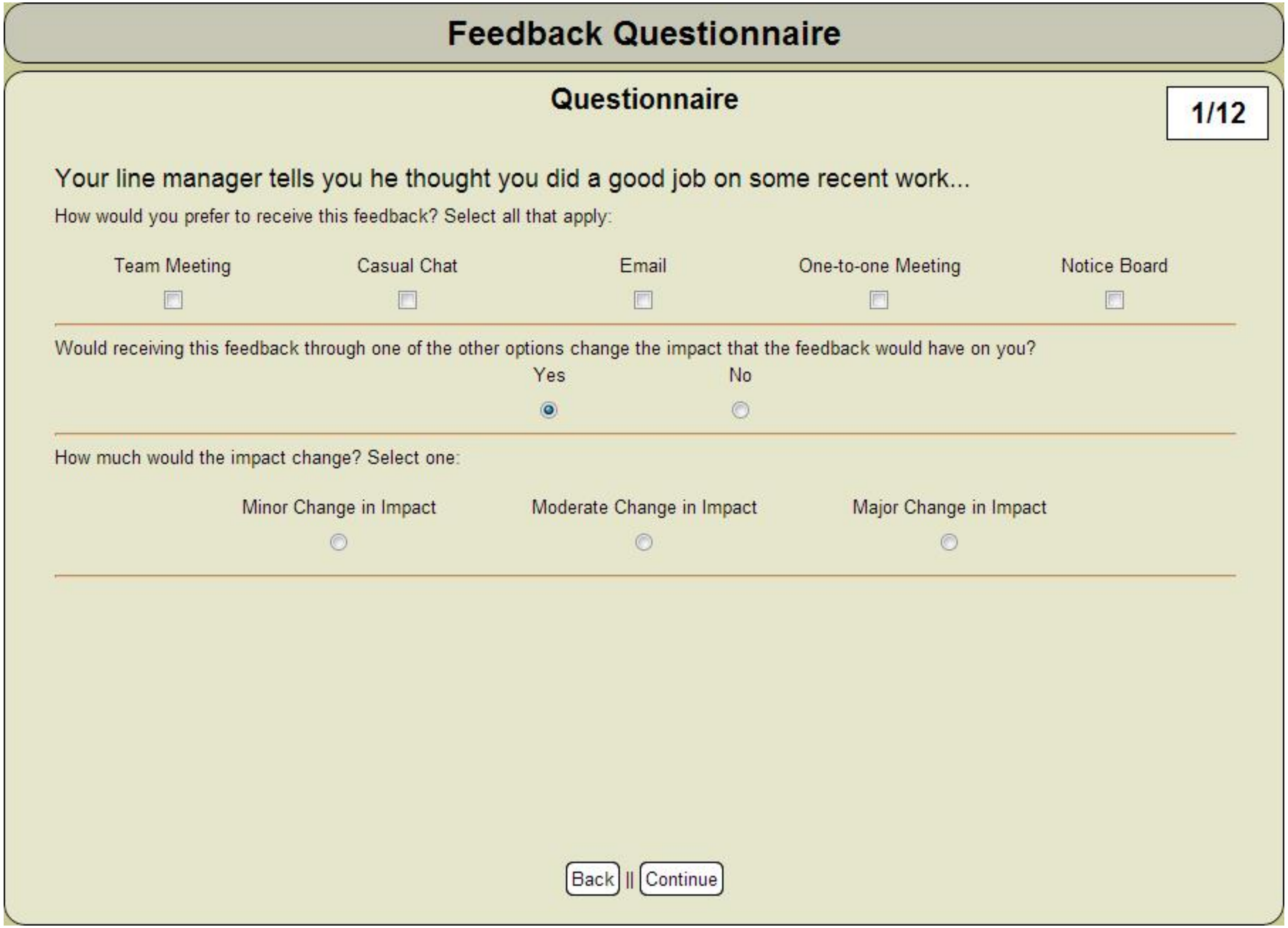
Which of the following types of development methods have you had experience with? Select all that apply:	Agile Methods □	Waterfall Model □	Spiral Model □	Iterative Development □	Other □

Which of the following tasks are typical of your current job? Select all that apply:	Requirements Gathering □	Systems Design □	Writing Code □	Systems Testing □	Systems Maintenance □	Other □

Including software engineers and other colleagues, what size development team are you typically in? Select One ▾

How many other software engineers are typically in your development team? Select One ▾

The following screens will present to you a scenario in which you receive feedback, and asks for your reactions. Please respond to each scenario with how you believe you would react if receiving the feedback.



Feedback Questionnaire

Questionnaire

1/12

Your line manager tells you he thought you did a good job on some recent work...

How would you prefer to receive this feedback? Select all that apply:

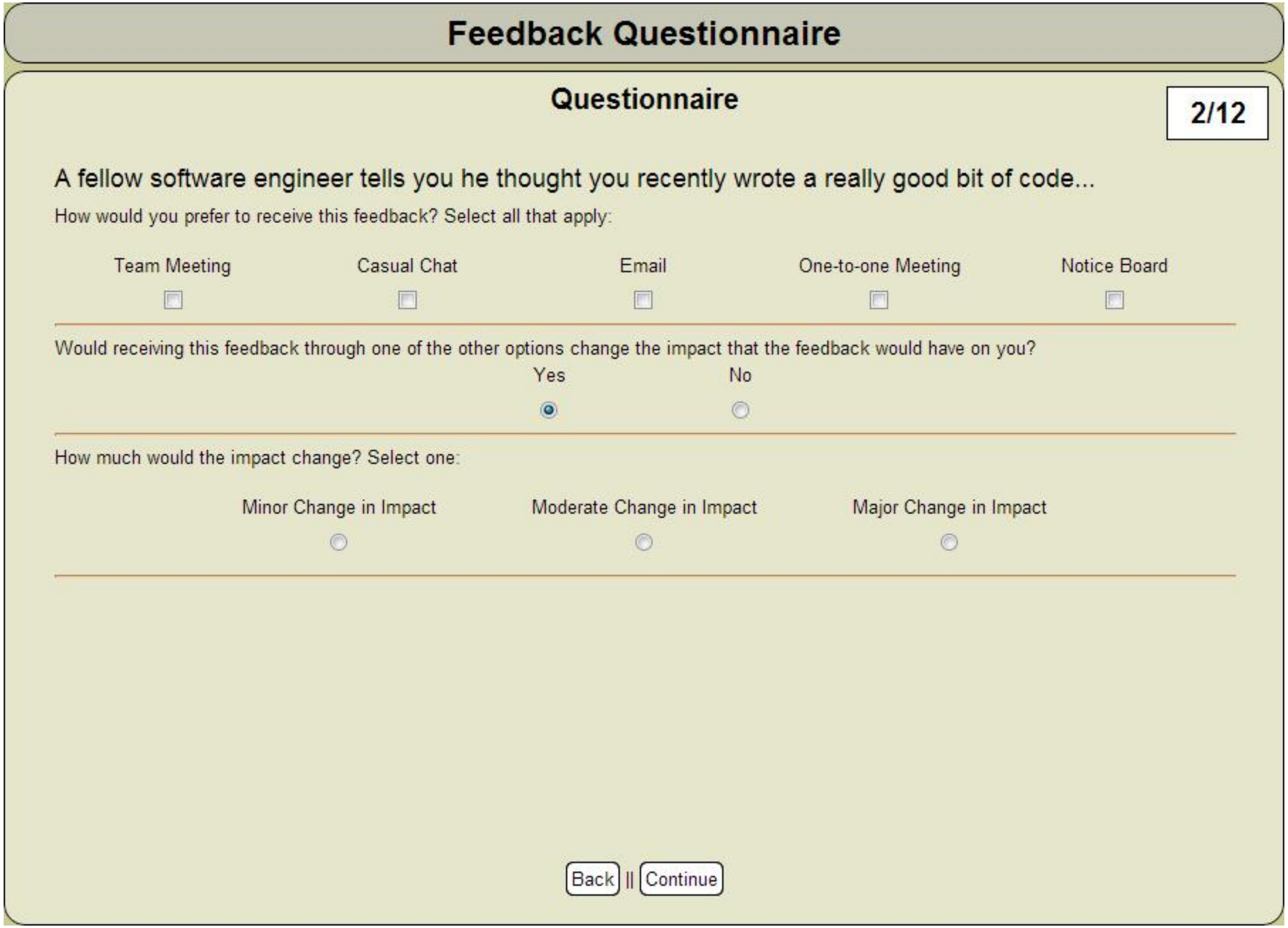
Team Meeting	Casual Chat	Email	One-to-one Meeting	Notice Board
□	□	□	□	□

Would receiving this feedback through one of the other options change the impact that the feedback would have on you?

Yes	No
●	○

How much would the impact change? Select one:

Minor Change in Impact	Moderate Change in Impact	Major Change in Impact
○	○	○



Feedback Questionnaire

Questionnaire

2/12

A fellow software engineer tells you he thought you recently wrote a really good bit of code...

How would you prefer to receive this feedback? Select all that apply:

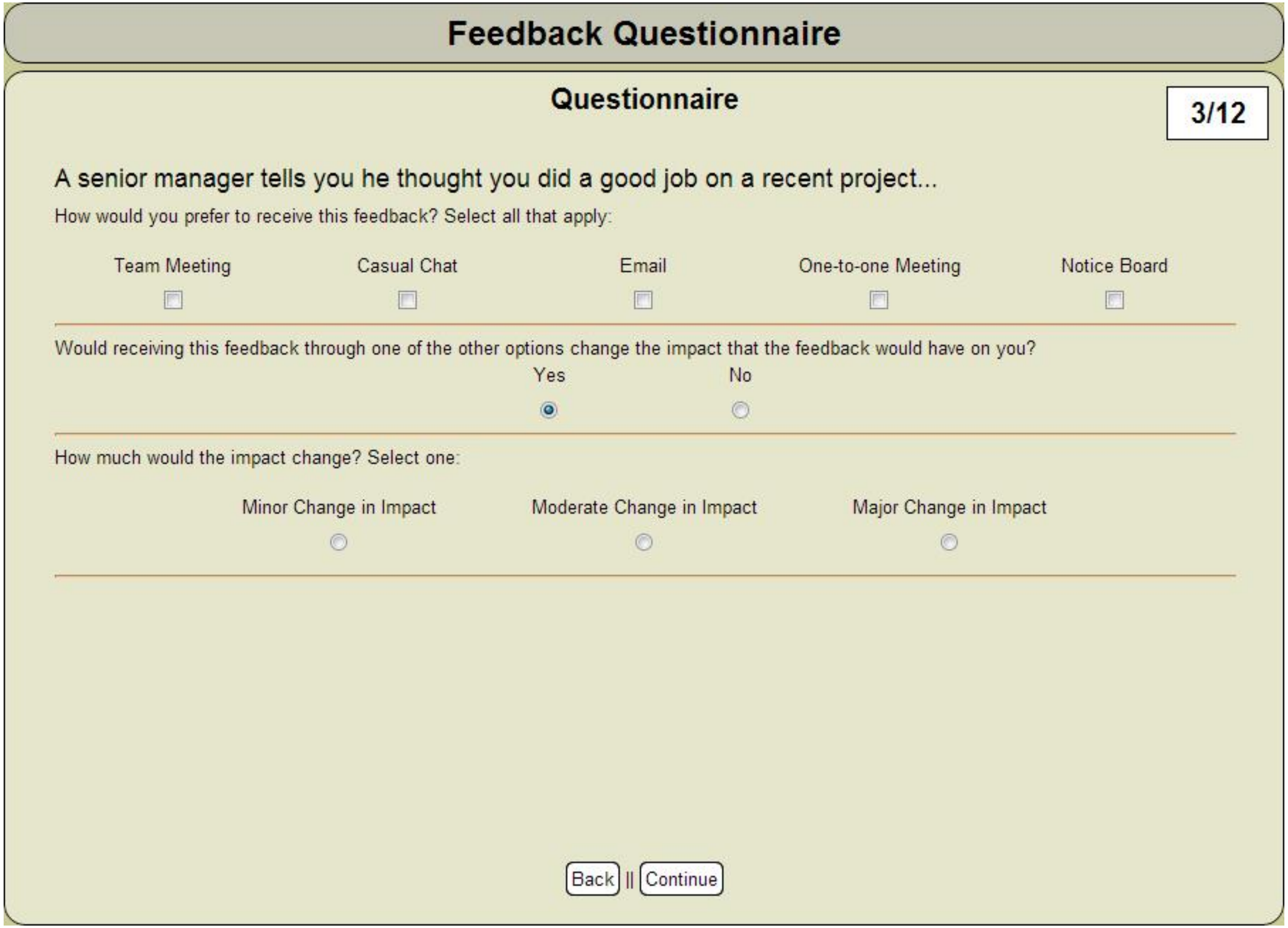
Team Meeting	Casual Chat	Email	One-to-one Meeting	Notice Board
□	□	□	□	□

Would receiving this feedback through one of the other options change the impact that the feedback would have on you?

Yes	No
●	○

How much would the impact change? Select one:

Minor Change in Impact	Moderate Change in Impact	Major Change in Impact
○	○	○



Feedback Questionnaire

Questionnaire

3/12

A senior manager tells you he thought you did a good job on a recent project...

How would you prefer to receive this feedback? Select all that apply:

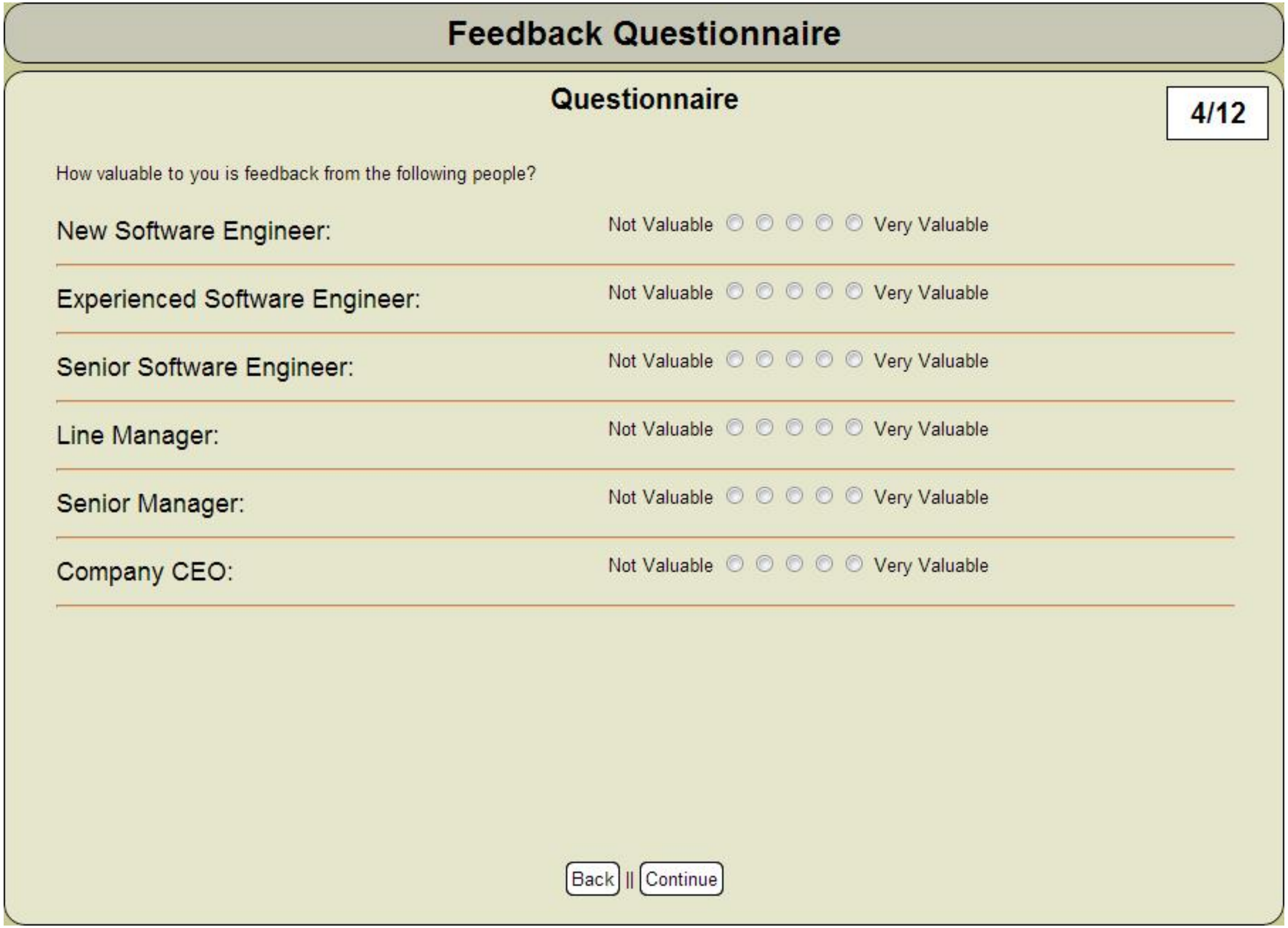
Team Meeting	Casual Chat	Email	One-to-one Meeting	Notice Board
□	□	□	□	□

Would receiving this feedback through one of the other options change the impact that the feedback would have on you?

Yes	No
●	○

How much would the impact change? Select one:

Minor Change in Impact	Moderate Change in Impact	Major Change in Impact
○	○	○



Feedback Questionnaire

Questionnaire

4/12

How valuable to you is feedback from the following people?

New Software Engineer: Not Valuable ○ ○ ○ ○ ○ Very Valuable

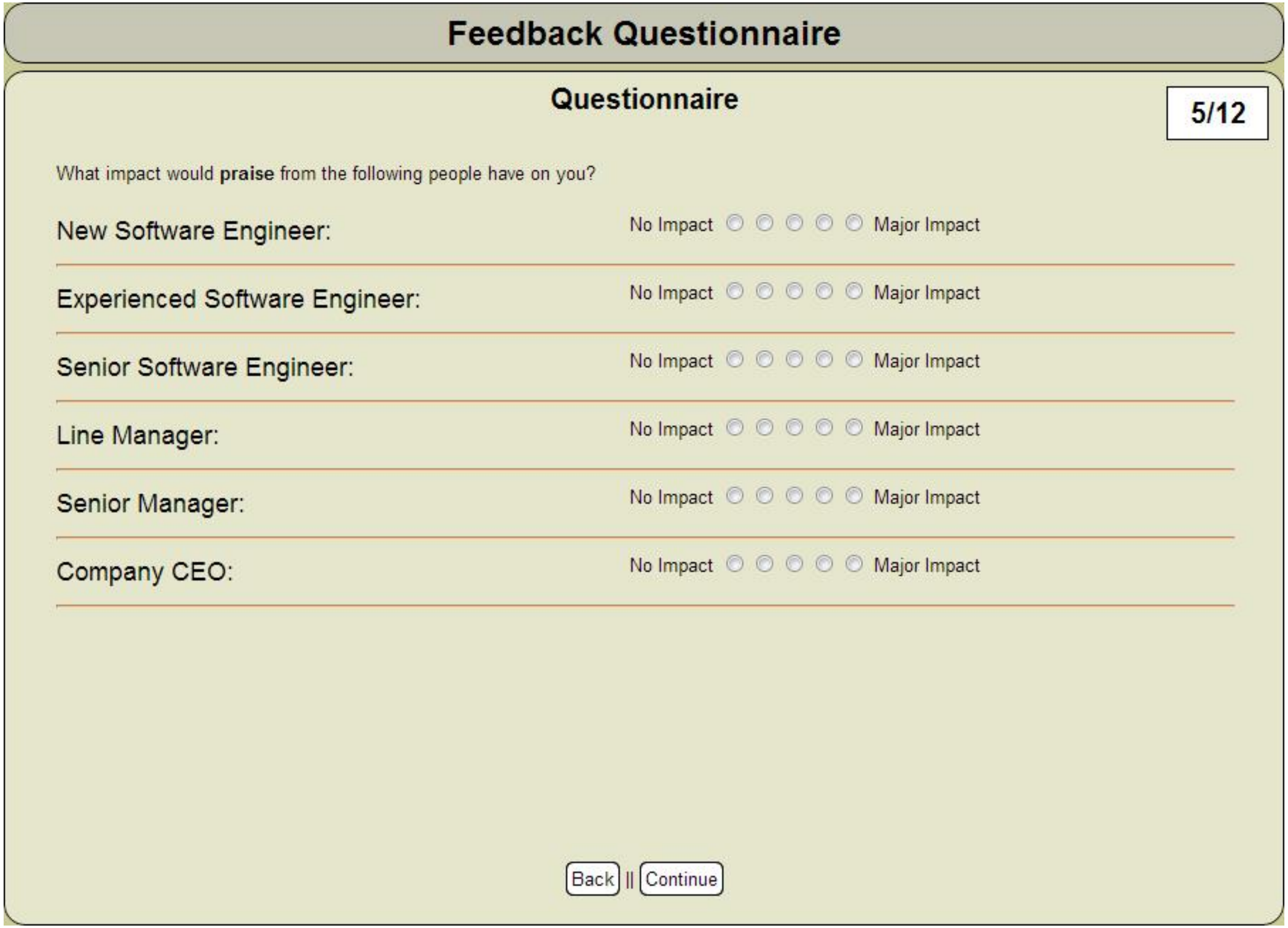
Experienced Software Engineer: Not Valuable ○ ○ ○ ○ ○ Very Valuable

Senior Software Engineer: Not Valuable ○ ○ ○ ○ ○ Very Valuable

Line Manager: Not Valuable ○ ○ ○ ○ ○ Very Valuable

Senior Manager: Not Valuable ○ ○ ○ ○ ○ Very Valuable

Company CEO: Not Valuable ○ ○ ○ ○ ○ Very Valuable



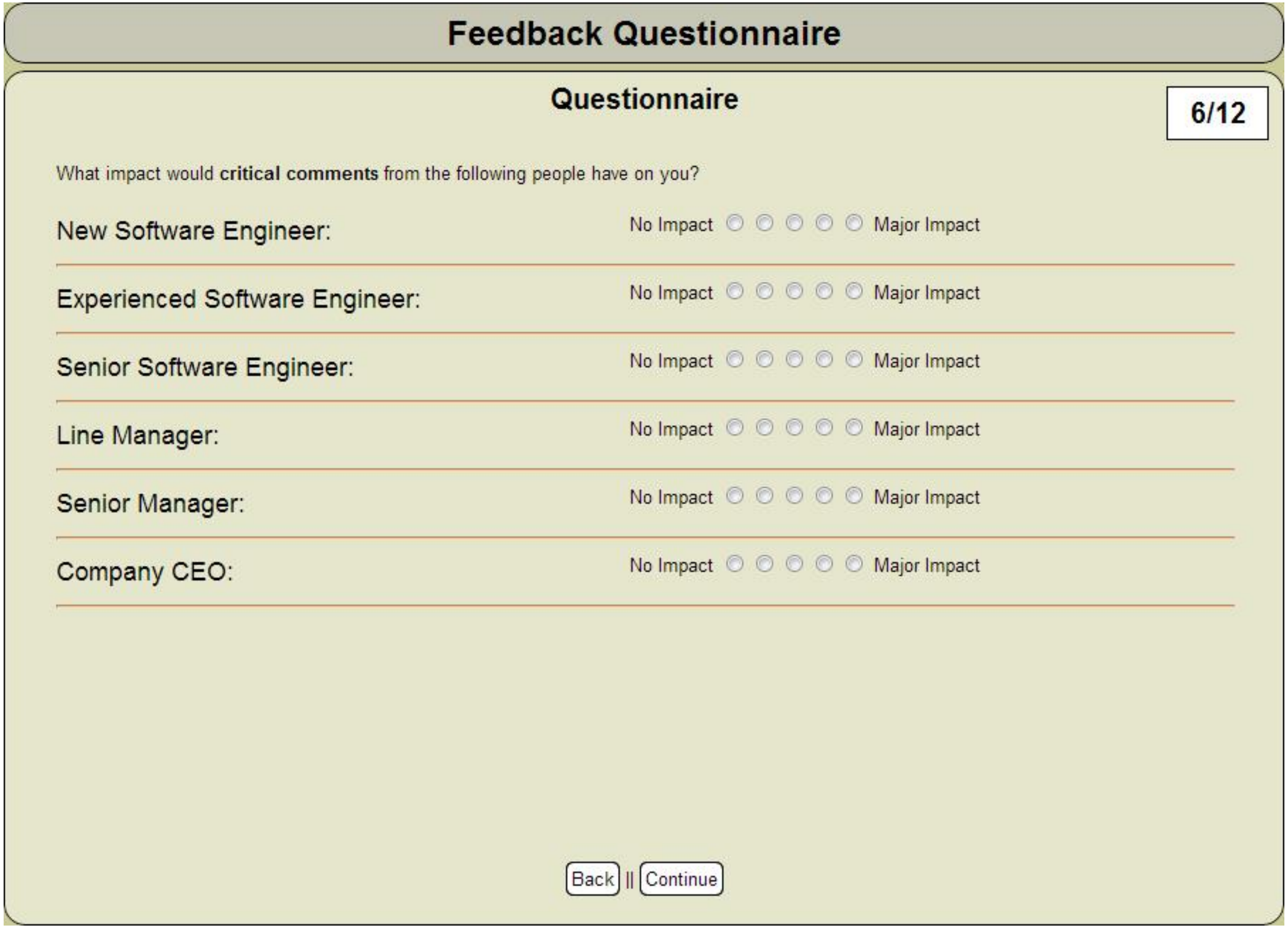
Feedback Questionnaire

Questionnaire

5/12

What impact would **praise** from the following people have on you?

New Software Engineer:	No Impact ○ ○ ○ ○ ○ Major Impact
Experienced Software Engineer:	No Impact ○ ○ ○ ○ ○ Major Impact
Senior Software Engineer:	No Impact ○ ○ ○ ○ ○ Major Impact
Line Manager:	No Impact ○ ○ ○ ○ ○ Major Impact
Senior Manager:	No Impact ○ ○ ○ ○ ○ Major Impact
Company CEO:	No Impact ○ ○ ○ ○ ○ Major Impact



Feedback Questionnaire

Questionnaire

6/12

What impact would **critical comments** from the following people have on you?

New Software Engineer: No Impact ○ ○ ○ ○ ○ Major Impact

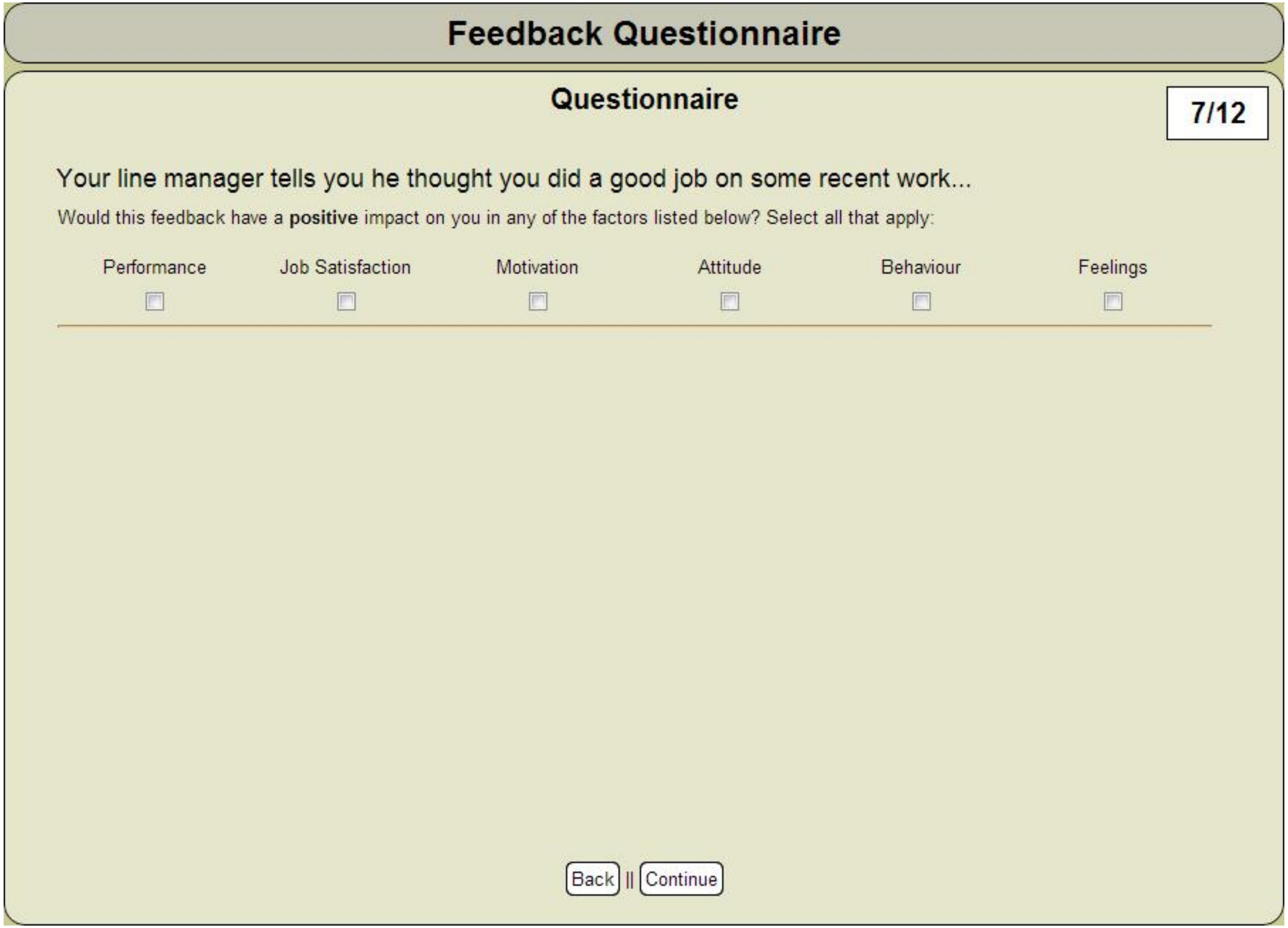
Experienced Software Engineer: No Impact ○ ○ ○ ○ ○ Major Impact

Senior Software Engineer: No Impact ○ ○ ○ ○ ○ Major Impact

Line Manager: No Impact ○ ○ ○ ○ ○ Major Impact

Senior Manager: No Impact ○ ○ ○ ○ ○ Major Impact

Company CEO: No Impact ○ ○ ○ ○ ○ Major Impact



Feedback Questionnaire

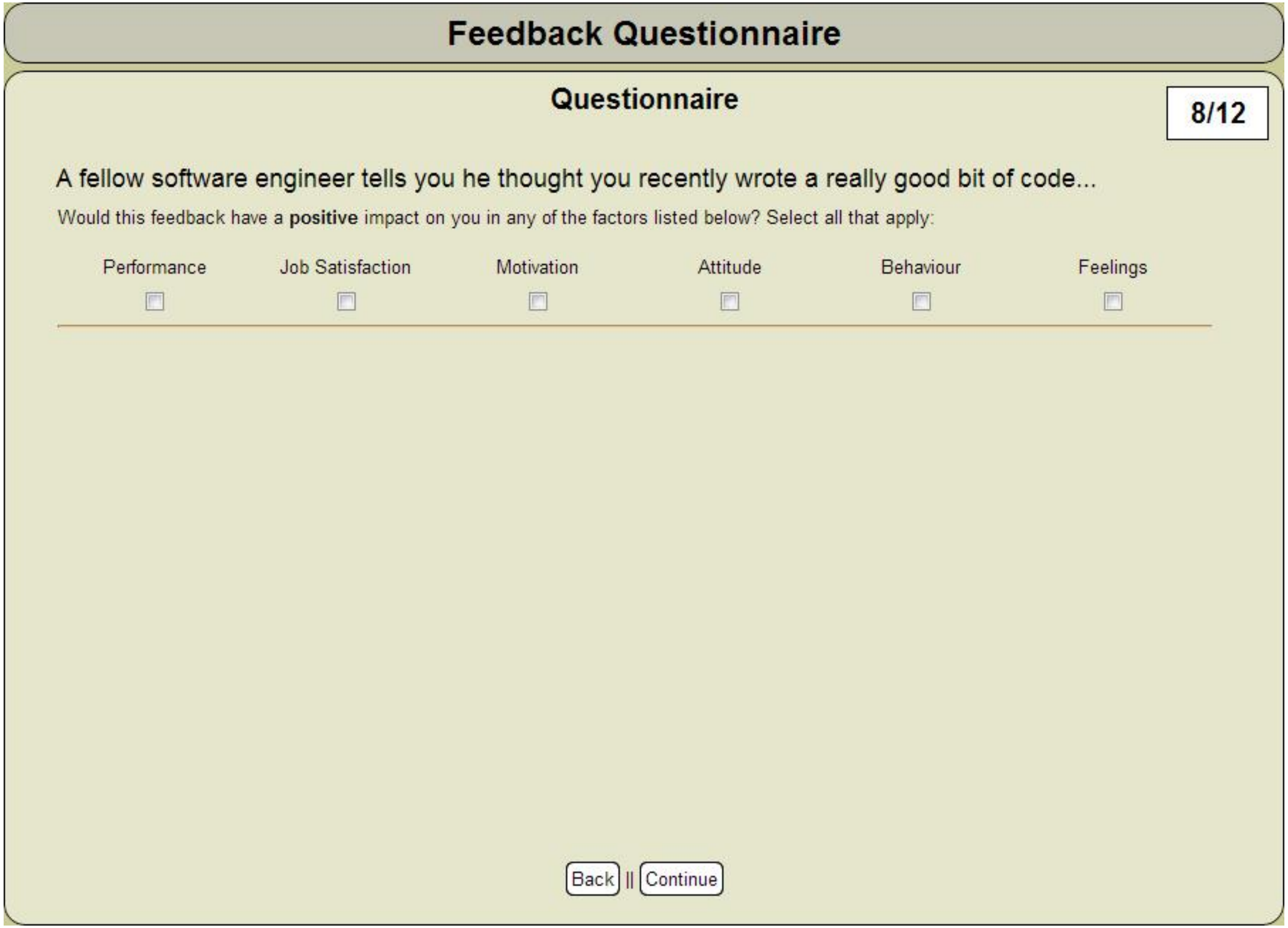
Questionnaire

7/12

Your line manager tells you he thought you did a good job on some recent work...

Would this feedback have a **positive** impact on you in any of the factors listed below? Select all that apply:

Performance	Job Satisfaction	Motivation	Attitude	Behaviour	Feelings
□	□	□	□	□	□



Feedback Questionnaire

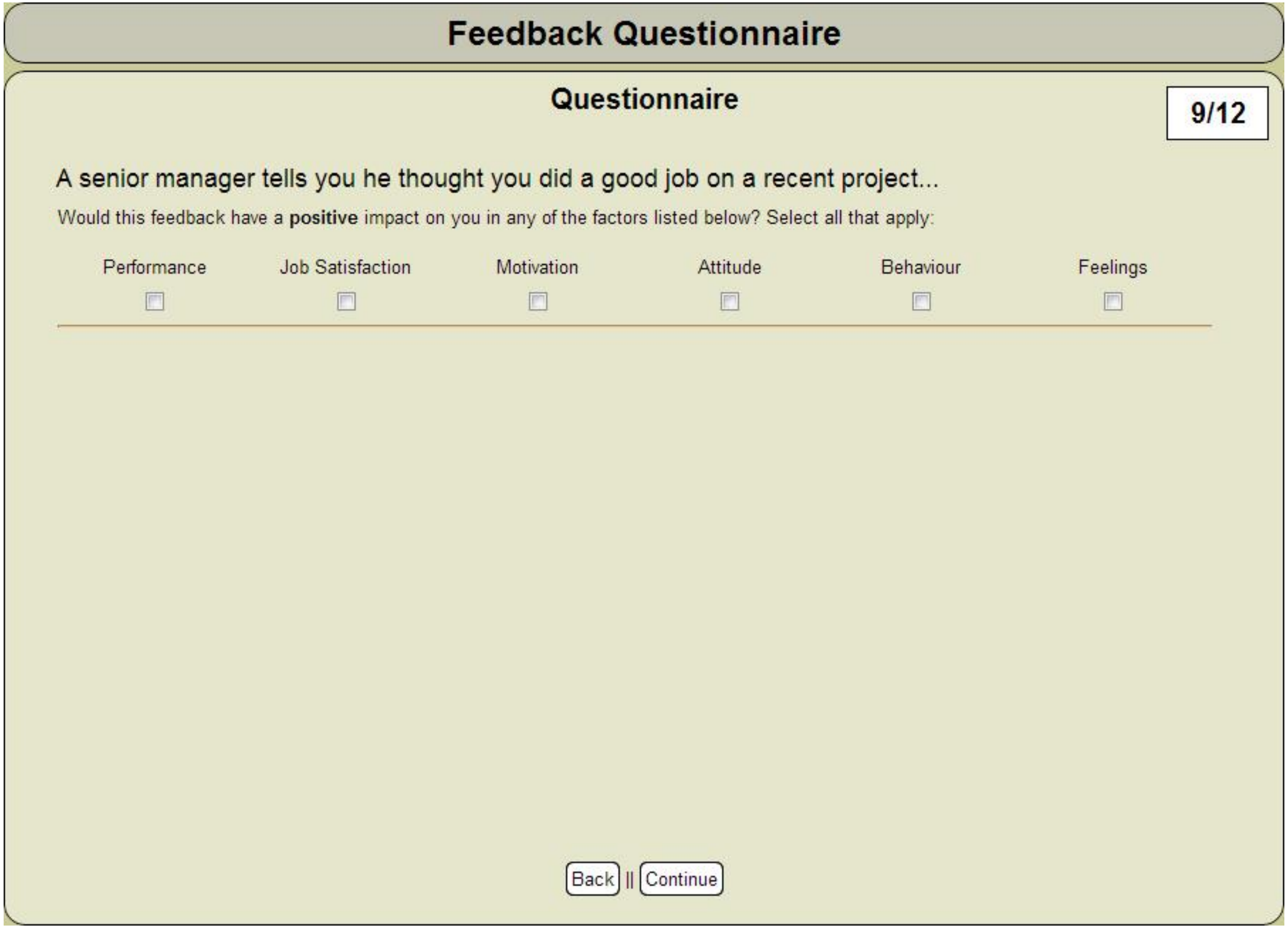
Questionnaire

8/12

A fellow software engineer tells you he thought you recently wrote a really good bit of code...

Would this feedback have a **positive** impact on you in any of the factors listed below? Select all that apply:

Performance	Job Satisfaction	Motivation	Attitude	Behaviour	Feelings
□	□	□	□	□	□



Feedback Questionnaire

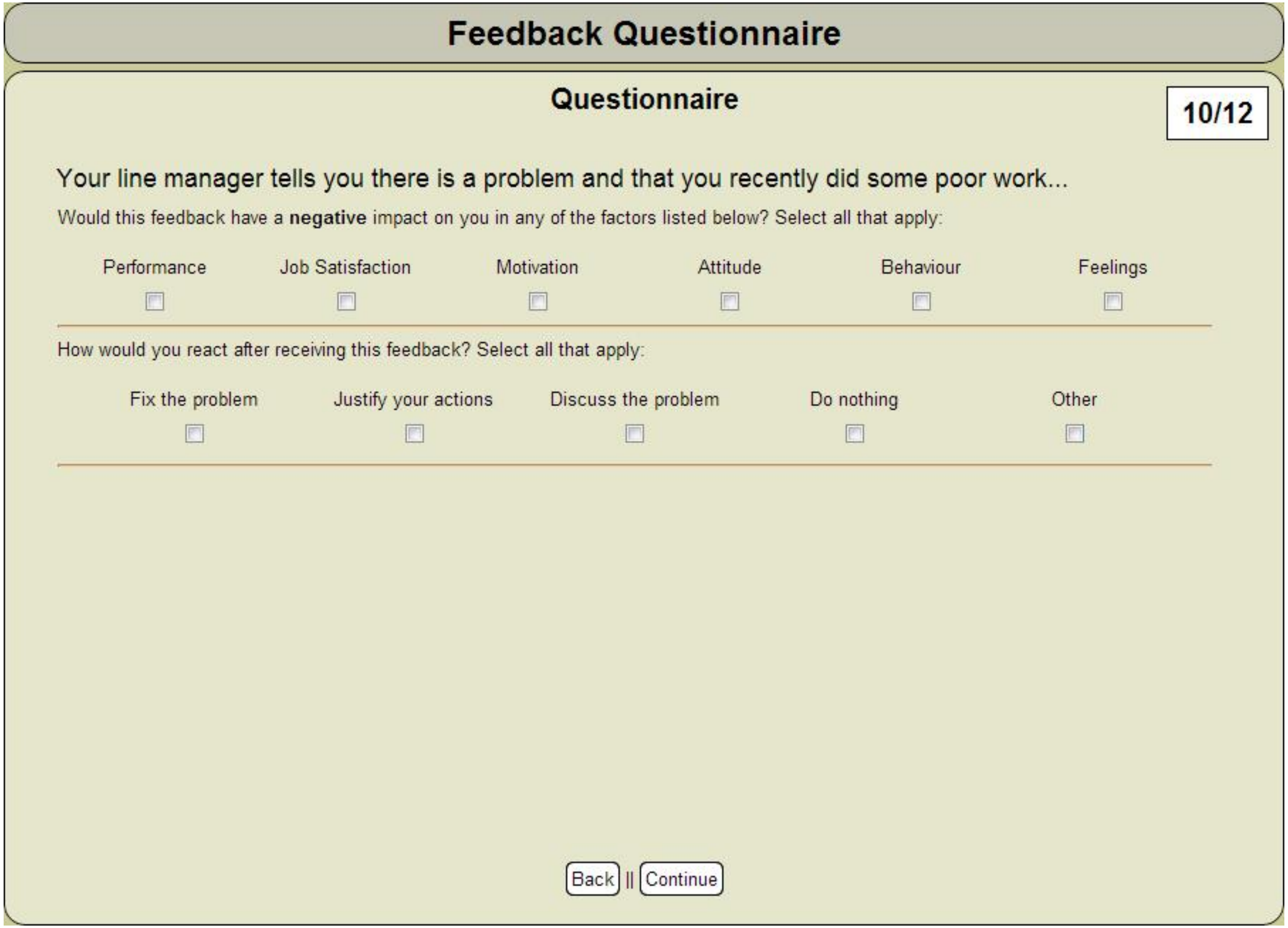
Questionnaire

9/12

A senior manager tells you he thought you did a good job on a recent project...

Would this feedback have a **positive** impact on you in any of the factors listed below? Select all that apply:

Performance	Job Satisfaction	Motivation	Attitude	Behaviour	Feelings
□	□	□	□	□	□



Feedback Questionnaire

Questionnaire

10/12

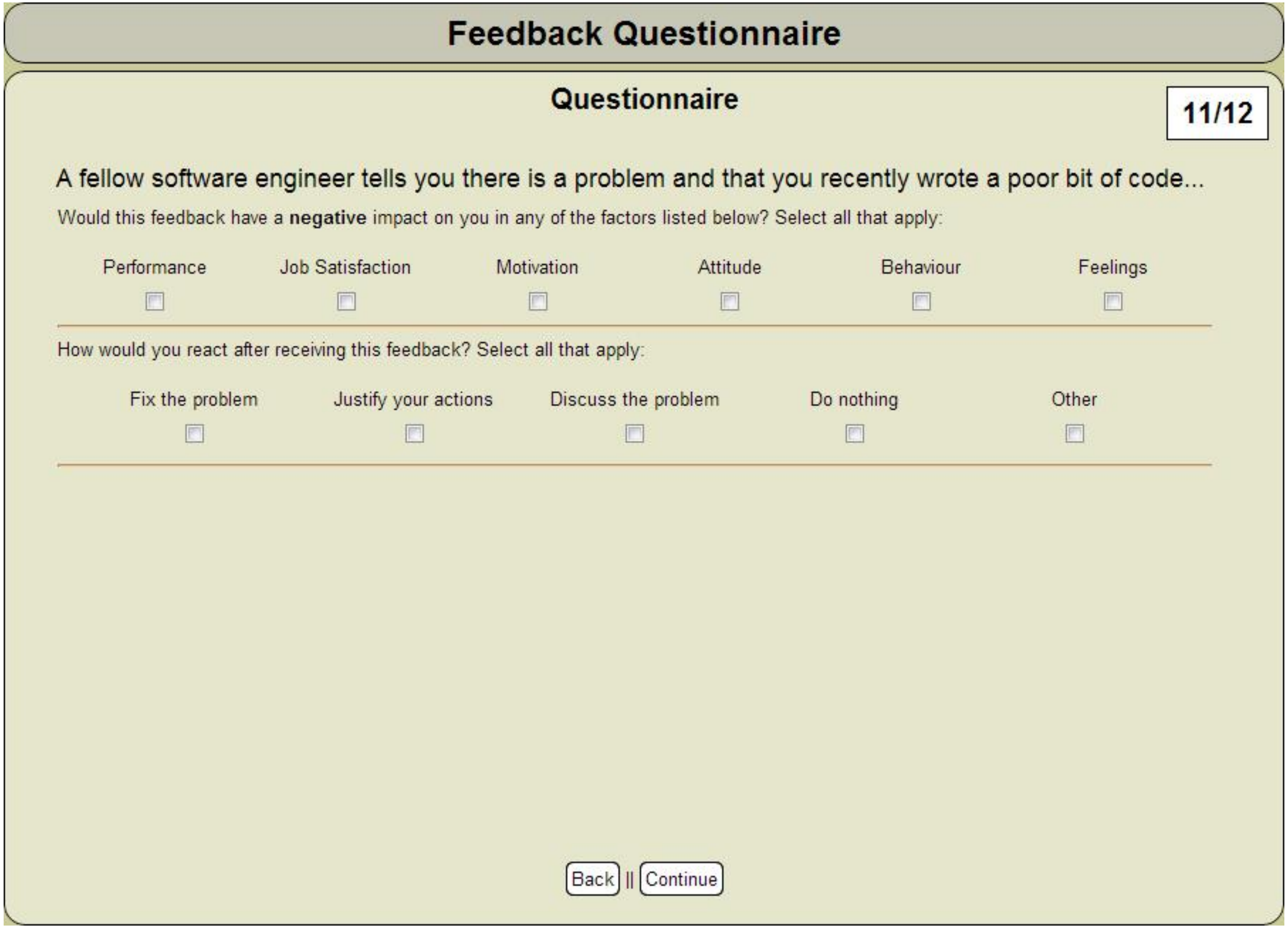
Your line manager tells you there is a problem and that you recently did some poor work...

Would this feedback have a **negative** impact on you in any of the factors listed below? Select all that apply:

Performance	Job Satisfaction	Motivation	Attitude	Behaviour	Feelings
□	□	□	□	□	□

How would you react after receiving this feedback? Select all that apply:

Fix the problem	Justify your actions	Discuss the problem	Do nothing	Other
□	□	□	□	□



Questionnaire

11/12

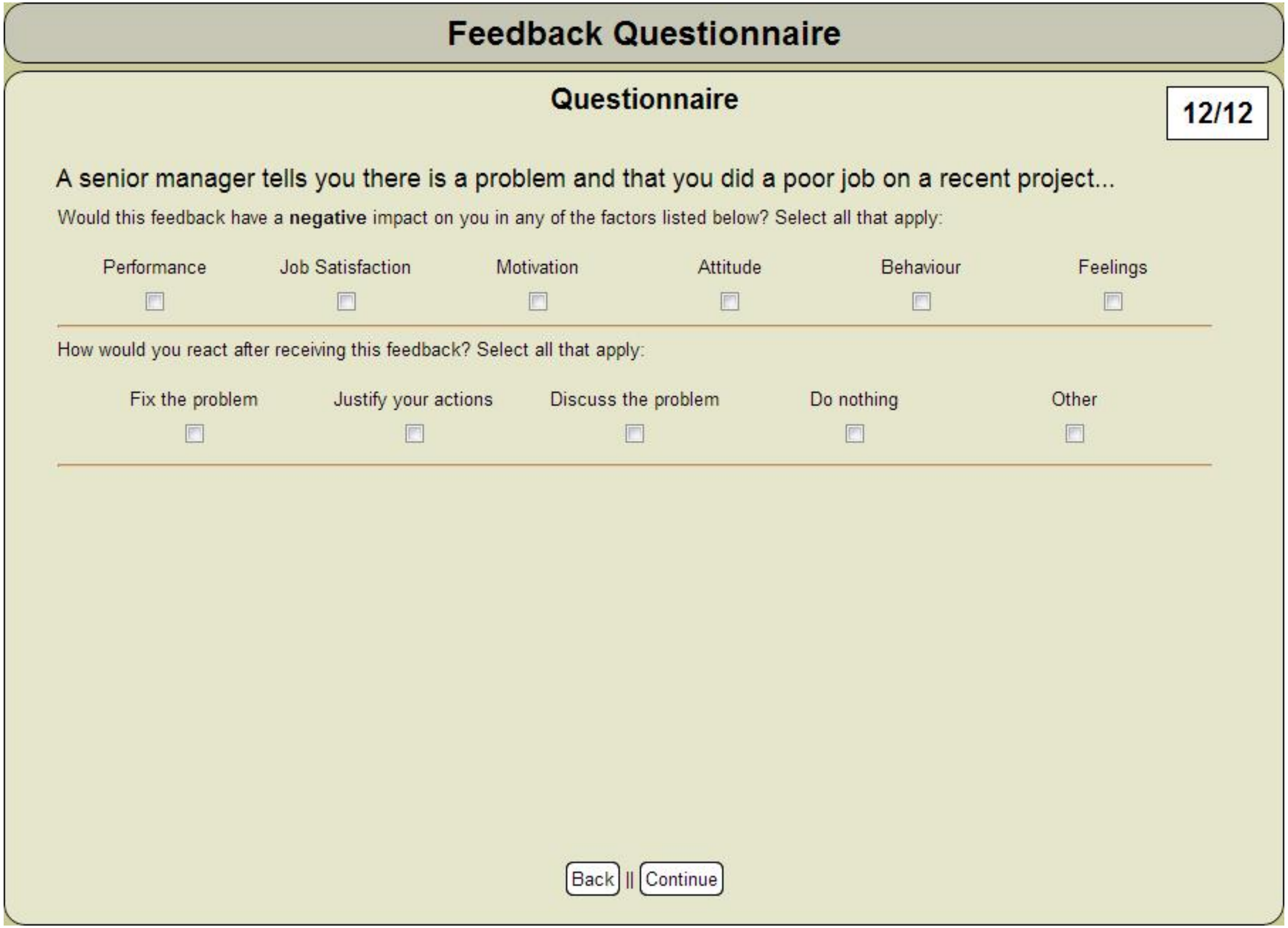
A fellow software engineer tells you there is a problem and that you recently wrote a poor bit of code...

Would this feedback have a **negative** impact on you in any of the factors listed below? Select all that apply:

Performance	Job Satisfaction	Motivation	Attitude	Behaviour	Feelings
□	□	□	□	□	□

How would you react after receiving this feedback? Select all that apply:

Fix the problem	Justify your actions	Discuss the problem	Do nothing	Other
□	□	□	□	□



Feedback Questionnaire

Questionnaire

12/12

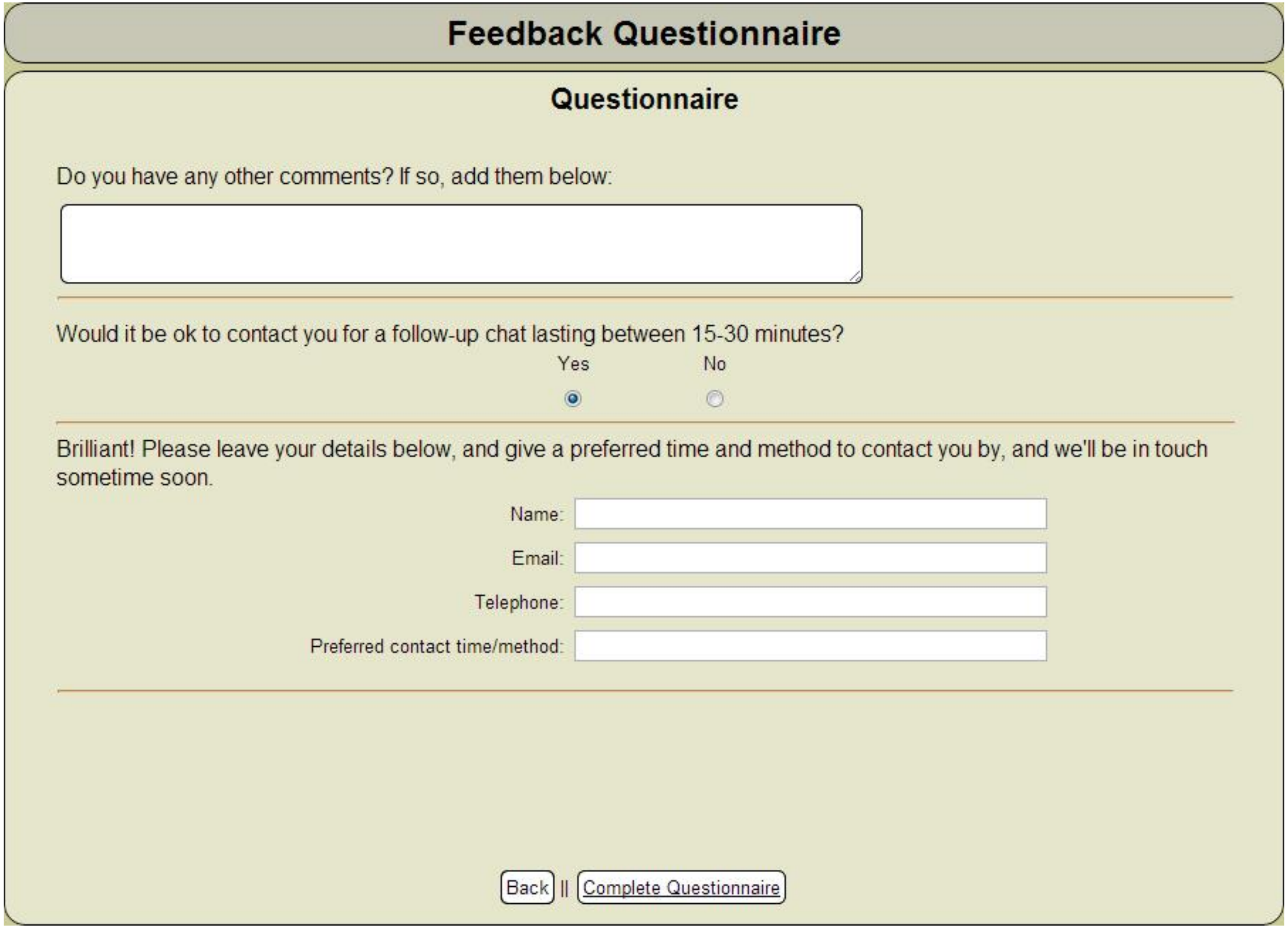
A senior manager tells you there is a problem and that you did a poor job on a recent project...

Would this feedback have a **negative** impact on you in any of the factors listed below? Select all that apply:

Performance	Job Satisfaction	Motivation	Attitude	Behaviour	Feelings
□	□	□	□	□	□

How would you react after receiving this feedback? Select all that apply:

Fix the problem	Justify your actions	Discuss the problem	Do nothing	Other
□	□	□	□	□



Feedback Questionnaire

Questionnaire

Do you have any other comments? If so, add them below:

Would it be ok to contact you for a follow-up chat lasting between 15-30 minutes?

Yes	No
●	○

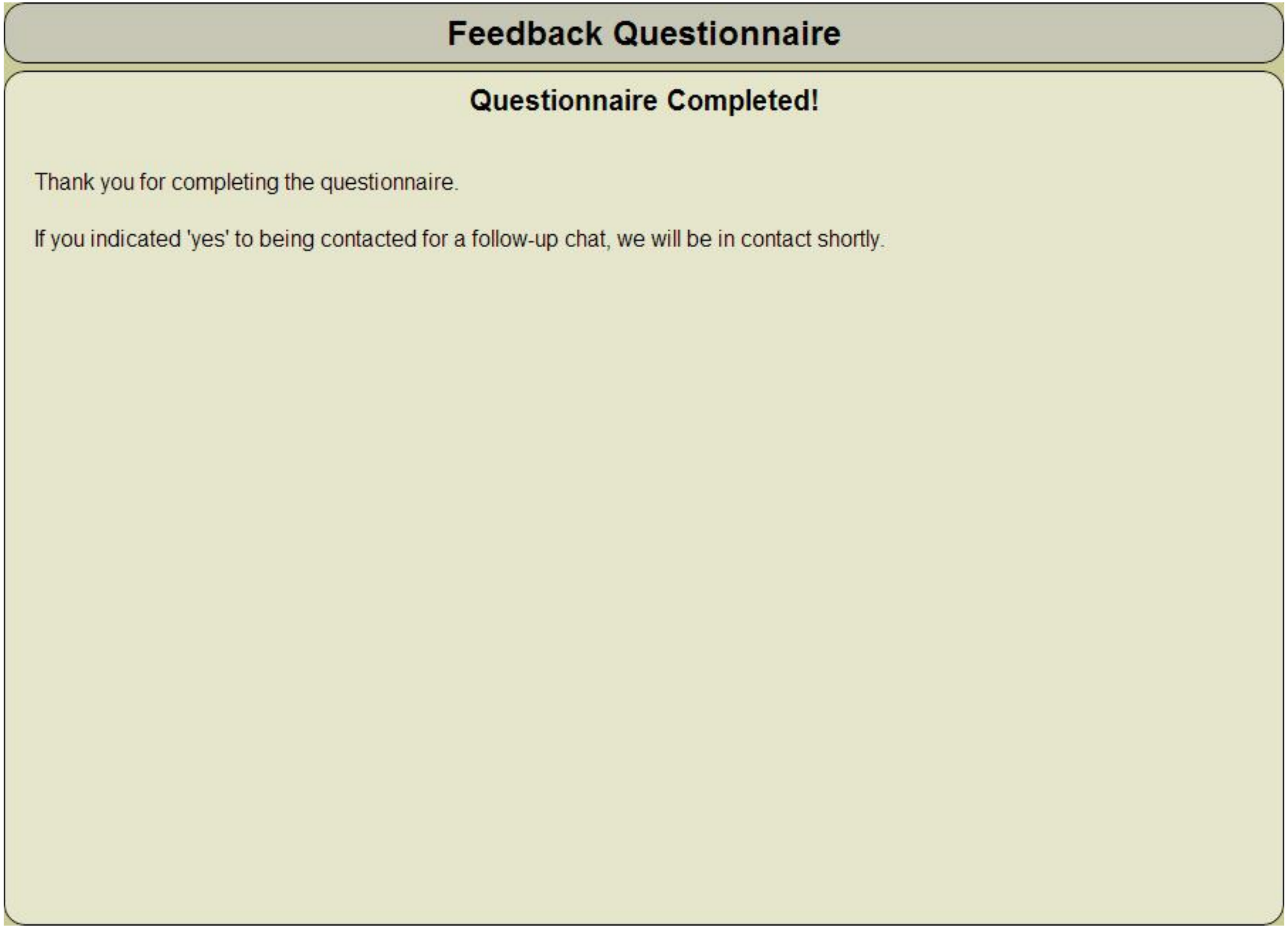
Brilliant! Please leave your details below, and give a preferred time and method to contact you by, and we'll be in touch sometime soon.

Name:

Email:

Telephone:

Preferred contact time/method:



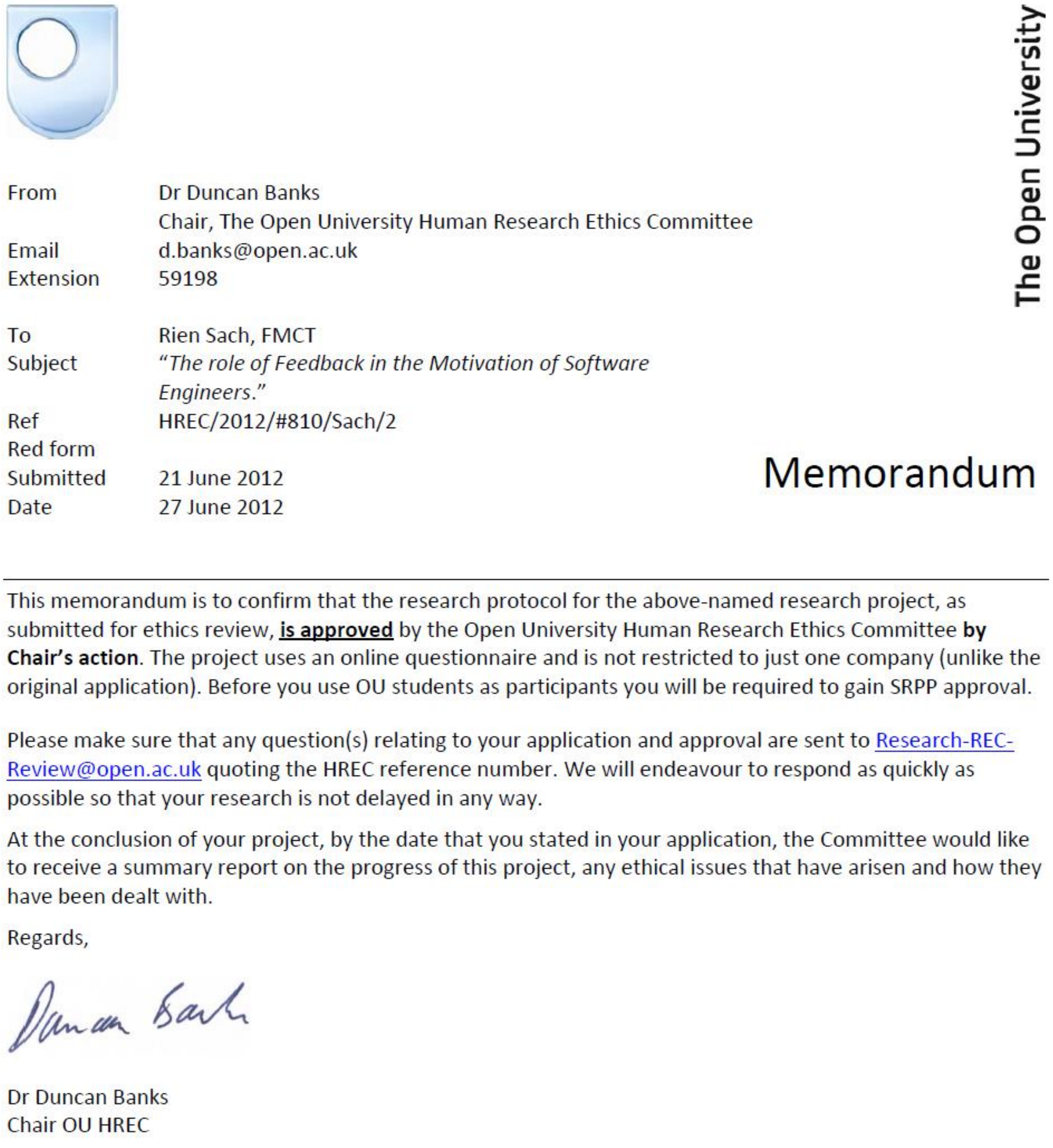
Feedback Questionnaire

Questionnaire Completed!

Thank you for completing the questionnaire.

If you indicated 'yes' to being contacted for a follow-up chat, we will be in contact shortly.

APPENDIX D SECTION 2 – THE EFFECT OF ‘SOURCE’ AND ‘MEDIUM’ FEEDBACK CHARACTERISTICS ETHICAL APPROVAL NOTIFICATION



The Open University

From	Dr Duncan Banks Chair, The Open University Human Research Ethics Committee
Email	d.banks@open.ac.uk
Extension	59198
To	Rien Sach, FMCT
Subject	*“The role of Feedback in the Motivation of Software Engineers.”*
Ref	HREC/2012/#810/Sach/2
Red form	
Submitted	21 June 2012
Date	27 June 2012

Memorandum

This memorandum is to confirm that the research protocol for the above-named research project, as submitted for ethics review, **is approved** by the Open University Human Research Ethics Committee **by Chair’s action**. The project uses an online questionnaire and is not restricted to just one company (unlike the original application). Before you use OU students as participants you will be required to gain SRPP approval.

Please make sure that any question(s) relating to your application and approval are sent to Research-REC-Review@open.ac.uk quoting the HREC reference number. We will endeavour to respond as quickly as possible so that your research is not delayed in any way.

At the conclusion of your project, by the date that you stated in your application, the Committee would like to receive a summary report on the progress of this project, any ethical issues that have arisen and how they have been dealt with.

Regards,

Duncan Banks

Dr Duncan Banks
Chair OU HREC

APPENDIX D SECTION 3 – THE EFFECT OF 'SOURCE' AND 'MEDIUM' FEEDBACK CHARACTERISTICS COMPARATIVE ANALYSIS FINDINGS

For the following analysis presentation, please use the attached key:
NSE = New Software Engineer
ESE = Experienced Software Engineer
SSE = Senior Software Engineering
LM = Line Manager
SM = Senior Manager
CEO = Company CEO
Please note, while full analysis of all possible permutations was taken, this section only shows the results of queries yielding at least 10% of the participants. There were more than 110,000 different permutations queried, and over 30,000 reporting at least 1 participant. Over 10,000 permutations had results for at least 10% of the participants. The results in this table are a snap-shot of important results identified from the reviewed 10,000+ permutations.

Feedback Value (Q4)	Praise Impact (Q5)	Critical Comments Impact (Q6)	Representation
Findings investigating 'greater than' relationships			
	praise from a senior software engineer *has greater impact on me than* praise from a new software engineer		75.16% (118/157)
		critical comments from a senior software engineer *have greater impact on me than* critical comments from a new software engineer	71.34% (112/157)
	praise from a senior software engineer *has greater impact on me than* praise from a new software engineer **AND** praise from an experienced software engineer *has greater impact on me than* praise from a new software engineer		67.52% (106/157)
		critical comments from a senior software engineer *have greater impact on me than* critical comments from a new software engineer **AND** critical comments from an	66.24% (104/157)

		experienced software engineer *have greater impact on me than* critical comments from a new software engineer	
feedback from an experienced software engineer *is of greater value to me than* feedback from a new software engineer			65.61% (103/157)
	praise from a senior software engineer *has greater impact on me than* praise from a new software engineer	critical comments from a senior software engineer *have greater impact on me than* critical comments from a new software engineer	62.42% (98/157)
feedback from a senior software engineer *is of greater value to me than* feedback from a new software engineer **AND** feedback from an experienced software engineer *is of greater value to me than* feedback from a new software engineer			61.78% (97/157)
feedback from a senior software engineer *is of greater value to me than* feedback from a new software engineer	praise from a senior software engineer *has greater impact on me than* praise from a new software engineer		57.96% (91/157)
feedback from a senior software engineer *is of greater value to me than* feedback from a new software engineer	praise from a senior software engineer *has greater impact on me than* praise from a new software engineer	critical comments from a senior software engineer *have greater impact on me than* critical comments from a new software engineer	48.41% (76/157)
feedback from an experienced software engineer *is of greater value to me than* feedback from a new software engineer	praise from an experienced software engineer *has greater impact on me than* praise from a new software engineer	critical comments from an experienced software engineer *have greater impact on me than* critical comments from a new software engineer	42.68% (67/157)
Findings investigating ‘less than’ relationships			
feedback from the company ceo *is of less value to me than* feedback from a line manager			40.76% (64/157)
feedback from the company ceo *is of less value to me than* feedback from a senior manager			32.48% (51/157)
feedback from a senior manager *is of less value to me than* feedback from a line manager			30.57% (48/157)

Findings investigating ‘greater than or equal to’ relationships

feedback from an experienced software engineer *is of equal or greater value to me than* feedback from a new software engineer	praise from an experienced software engineer *has equal or greater impact on me than* praise from a new software engineer		99.36% (156/157)
feedback from a senior software engineer *is of equal or greater value to me than* feedback from a new software engineer			98.73% (155/157)
	praise from a senior software engineer *has equal or greater impact on me than* praise from a new software engineer **AND** praise from an experienced software engineer *has equal or greater impact on me than* praise from a new software engineer		98.73% (155/157)
	praise from an experienced software engineer *has equal or greater impact on me than* praise from a new software engineer **AND** praise from a senior software engineer *has equal or greater impact on me than* praise from an experienced software engineer		97.45% (153/157)
feedback from a senior software engineer *is of equal or greater value to me than* feedback from a new software engineer **AND** feedback from an experienced software engineer *is of equal or greater value to me than* feedback from a new software engineer	praise from a senior software engineer *has equal or greater impact on me than* praise from a new software engineer **AND** praise from an experienced software engineer *has equal or greater impact on me than* praise from a new software engineer		97.45% (153/157)
		critical comments from a senior software engineer *have equal or greater impact on me than* critical comments from a new software engineer **AND** critical comments from an experienced software engineer *have equal or greater impact on me than* critical comments from a new software engineer	94.9% (149/157)
	praise from an experienced software engineer *has equal or greater impact on me than* praise from a new software engineer	critical comments from an experienced software engineer *have equal or greater impact on me than* critical comments from a new software engineer	94.9% (149/157)

feedback from an experienced software engineer *is of equal or greater value to me than* feedback from a new software engineer	praise from an experienced software engineer *has equal or greater impact on me than* praise from a new software engineer	critical comments from an experienced software engineer *have equal or greater impact on me than* critical comments from a new software engineer	94.9% (149/157)
	praise from the company ceo *has equal or greater impact on me than* praise from a senior manager		85.99% (135/157)
		critical comments from a senior manager *have equal or greater impact on me than* critical comments from a line manager	81.53% (128/157)
	praise from the company ceo *has equal or greater impact on me than* praise from a senior manager	critical comments from the company ceo *have equal or greater impact on me than* critical comments from a senior manager	80.25% (126/157)
	praise from the company ceo *has equal or greater impact on me than* praise from a line manager		77.07% (121/157)
	praise from the company ceo *has equal or greater impact on me than* praise from a line manager **AND** praise from the company ceo *has equal or greater impact on me than* praise from a senior manager		76.43% (120/157)
	praise from the company ceo *has equal or greater impact on me than* praise from a line manager **AND** praise from a senior manager *has equal or greater impact on me than* praise from a line manager **AND** praise from the company ceo *has equal or greater impact on me than* praise from a senior manager		73.89% (116/157)
		critical comments from the company ceo *have equal or greater impact on me than* critical comments from a line manager **AND** critical comments from the company ceo *have equal or greater impact on me than* critical comments from a senior manager **AND** critical comments from a senior manager *have equal or greater impact on me than* critical comments from a line manager	73.89% (116/157)

feedback from the company ceo *is of equal or greater value to me than* feedback from a senior manager	praise from the company ceo *has equal or greater impact on me than* praise from a senior manager	critical comments from the company ceo *have equal or greater impact on me than* critical comments from a senior manager	60.51% (95/157)
Findings investigating ‘equal to’ relationships			
feedback from a senior software engineer *is of equal value to me as* feedback from an experienced software engineer			77.71% (122/157)
		critical comments from the company ceo *have equal impact on me as* critical comments from a senior manager	76.43% (120/157)
	praise from the company ceo *has equal impact on me as* praise from a senior manager		67.52% (106/157)
feedback from a senior software engineer *is of equal value to me as* feedback from an experienced software engineer		critical comments from a senior software engineer *has equal impact on me as* critical comments from an experienced software engineer	61.78% (97/157)
feedback from a senior software engineer *is of equal value to me as* feedback from an experienced software engineer	praise from a senior software engineer *has equal impact on me as* praise from an experienced software engineer		61.15% (96/157)
		critical comments from a senior software engineer *has equal impact on me as* critical comments from an experienced software engineer **AND** critical comments from the company ceo *has equal impact on me as* critical comments from a senior manager	55.41% (87/157)
		critical comments from the company ceo *has equal impact on me as* critical comments from a line manager	49.04% (77/157)
feedback from the company ceo *is of equal value to me as* feedback from a senior manager		critical comments from the company ceo *has equal impact on me as* critical comments from a senior manager	48.41% (76/157)

APPENDIX D SECTION 4 – THE EFFECT OF 'SOURCE' AND 'MEDIUM' FEEDBACK CHARACTERISTICS CONFIDENCE INTERVAL RANGES

The following tables show the confidence interval ranges for figures presented in chapter 7, with a 90% confidence level.

FIGURE 7.1 CONFIDENCE INTERVALS

Feedback	Team Meeting	Casual Chat	Email	One-to-one Meeting	Notice Board
Line Manager	34%	75%	41%	52%	3%
Software Engineer	22%	90%	36%	18%	2%
Senior Manager	43%	51%	54%	40%	6%
Line Manager C.O. Range	6%	6%	7%	7%	2%
Software Engineer C.O. Range	6%	4%	6%	5%	2%
Senior Manager C.O. Range	7%	7%	7%	6%	3%

FIGURE 7.2 CONFIDENCE INTERVALS

Feedback	1 (not valuable)	2	3	4	5 (very valuable)	No Response
New Software Engineer	9%	23%	32%	18%	18%	1%
Experienced Software Engineer	0%	3%	12%	36%	49%	1%
Senior Software Engineer	1%	1%	8%	31%	57%	1%
Line Manager	2%	5%	23%	38%	32%	1%
Senior Manager	3%	10%	30%	31%	26%	1%
Company CEO	11%	14%	26%	20%	27%	1%
New Software Engineer C.O. Range	4%	6%	6%	5%	5%	1%
Experienced Software Engineer C.O. Range	0%	2%	4%	6%	7%	1%
Senior Software Engineer C.O. Range	1%	1%	4%	6%	7%	1%
Line Manager C.O. Range	2%	3%	6%	6%	6%	1%
Senior Manager C.O. Range	2%	4%	6%	6%	6%	1%
Company CEO C.O. Range	4%	5%	6%	5%	6%	1%

FIGURE 7.3 CONFIDENCE INTERVALS

Feedback	1 (no impact)	2	3	4	5 (major impact)	No Response
New Software Engineer	13%	29%	37%	13%	8%	1%
Experienced Software Engineer	3%	3%	27%	41%	25%	1%
Senior Software Engineer	1%	2%	17%	41%	36%	2%
Line Manager	2%	5%	27%	41%	24%	1%
Senior Manager	2%	8%	25%	36%	28%	1%
Company CEO	5%	8%	20%	25%	39%	2%
New Software Engineer C.O. Range	4%	6%	6%	4%	4%	1%
Experienced Software Engineer C.O. Range	2%	2%	6%	7%	6%	1%
Senior Software Engineer C.O. Range	1%	2%	5%	7%	6%	2%
Line Manager C.O. Range	2%	3%	6%	7%	6%	1%
Senior Manager C.O. Range	2%	4%	6%	6%	6%	1%
Company CEO C.O. Range	3%	4%	5%	6%	6%	2%

FIGURE 7.4 CONFIDENCE INTERVALS

Feedback	1 (no impact)	2	3	4	5 (major impact)	No Response
New Software Engineer	15%	25%	32%	18%	8%	2%
Experienced Software Engineer	1%	4%	27%	38%	28%	2%
Senior Software Engineer	1%	4%	18%	41%	34%	2%
Line Manager	3%	8%	26%	35%	26%	3%
Senior Manager	4%	10%	22%	31%	31%	2%
Company CEO	8%	9%	24%	20%	38%	2%
New Software Engineer C.O. Range	5%	6%	6%	5%	4%	2%
Experienced Software Engineer C.O. Range	1%	3%	6%	6%	6%	2%
Senior Software Engineer C.O. Range	1%	3%	5%	7%	6%	2%
Line Manager C.O. Range	2%	4%	6%	6%	6%	2%
Senior Manager C.O. Range	3%	4%	6%	6%	6%	2%
Company CEO C.O. Range	4%	4%	6%	5%	6%	2%

FIGURE 7.5 CONFIDENCE INTERVALS

Feedback	Performance	Job Satisfaction	Motivation	Attitude	Behaviour	Feelings	No Response
Line Manager	41%	90%	81%	47%	24%	73%	1%
Software Engineer	27%	66%	57%	42%	20%	75%	1%
Senior Manager	32%	82%	69%	38%	17%	64%	3%
Line Manager C.O. Range	7%	4%	5%	7%	6%	6%	1%
Software Engineer C.O. Range	6%	6%	7%	7%	5%	6%	1%
Senior Manager C.O. Range	6%	5%	6%	6%	5%	6%	3%

FIGURE 7.6 CONFIDENCE INTERVALS

Feedback	Performance	Job Satisfaction	Motivation	Attitude	Behaviour	Feelings	No Response
Line Manager	15%	41%	39%	24%	18%	69%	8%
Software Engineer	6%	30%	25%	23%	11%	64%	15%
Senior Manager	15%	52%	45%	25%	15%	63%	10%
Line Manager C.O. Range	5%	7%	6%	6%	5%	6%	3%
Software Engineer C.O. Range	3%	6%	6%	6%	4%	6%	5%
Senior Manager C.O. Range	5%	7%	7%	6%	5%	6%	4%

FIGURE 7.7 CONFIDENCE INTERVALS

Feedback	Fix the problem	Justify your actions	Discuss the problem	Do Nothing	Other	No Response
Line Manager	85%	38%	89%	1%	6%	1%
Software Engineer	79%	38%	87%	3%	4%	1%
Senior Manager	78%	36%	80%	4%	6%	3%
Line Manager C.O. Range	5%	6%	4%	1%	3%	1%
Software Engineer C.O. Range	5%	6%	5%	2%	3%	1%
Senior Manager C.O. Range	6%	6%	5%	3%	3%	3%

FIGURE 7.8 CONFIDENCE INTERVALS

Key	Percentage	C.O. Range
ESE >= NSE	99.36%	1.05
SSE = ESE	77.71%	5.51
LM = SM	56.69%	6.56
CEO = SM	55.41%	6.59
ESE > SM	52.87%	6.61
SSE > SM	52.87%	6.61
SSE > LM	43.95%	6.58
SSE = LM	42.68%	6.55
ESE > LM	42.04%	6.54
CEO < LM	40.76%	6.51
ESE = LM	39.49%	6.48
SSE = SM	35.03%	6.32
LM > SM	30.57%	6.1
ESE = SM	28.03%	5.95
SSE >= NSE & ESE >= NSE	98.09%	1.81
SSE > NSE & ESE > NSE	61.78%	6.44
SSE > SM & ESE > SM	49.58%	6.62

FIGURE 7.9 CONFIDENCE INTERVALS

Key	Percentage	C.O. Range
ESE >= NSE	99.36%	1.05
CEO >= SM	85.99%	4.6
SSE > NSE	75.16%	5.72
CEO = SM	67.52%	6.2
SSE > SM	42.04%	6.54
SSE = LM	40.76%	6.51
ESE = LM	40.13%	6.49
SSE > LM	38.85%	6.46
ESE = SM	36.31%	6.37
SSE = SM	33.76%	6.26
ESE > SM	31.21%	6.14
ESE > LM	28.66%	5.99
SSE >= NSE & ESE >= NSE	98.73%	1.49
CEO >= LM & SM >= LM & CEO >= SM	73.89%	5.82
SSE > NSE & ESE > NSE	67.52%	6.2

FIGURE 7.10 CONFIDENCE INTERVALS

Key	**Percentage**	**C.O. Range**
ESE >= NSE	95.54%	2.73
CEO >= SM	88.54%	4.22
SM >= LM	81.53%	5.14
CEO = SM	76.43%	5.62
SSE = ESE	72.61%	5.91
SSE > NSE	71.34%	5.99
LM = SSE	49.04%	6.62
SSE = LM	49.04%	6.62
LM = ESE	46.50%	6.61
ESE = LM	46.50%	6.61
SSE > SM	40.76%	6.51
ESE > SM	35.76%	6.35
SSE = SM	33.76%	6.26
ESE = SM	33.76%	6.26
SSE > LM	33.12%	6.24
ESE > LM	27.39%	5.91
SSE >= NSE & ESE >= NSE	94.90%	2.91
SSE > NSE & ESE > NSE	66.24%	6.26

FIGURE 7.11 CONFIDENCE INTERVALS

Feedback	**Performance**	**Job Satisfaction**	**Motivation**	**Attitude**	**Behaviour**	**Feelings**	**No Response**
Line Manager	38%	90%	79%	50%	16%	71%	20%
Software Engineer	34%	72%	62%	47%	19%	67%	0%
Senior Manager	21%	81%	59%	36%	14%	64%	3%
Line Manager C.O. Range	7%	7%	9%	11%	8%	10%	8%
Software Engineer C.O. Range	10%	10%	11%	11%	9%	10%	0%
Senior Manager C.O. Range	9%	9%	11%	11%	8%	11%	3%

FIGURE 7.12 CONFIDENCE INTERVALS

Feedback	Performance	Job Satisfaction	Motivation	Attitude	Behaviour	Feelings	No Response
Line Manager	21%	41%	47%	31%	19%	69%	5%
Software Engineer	3%	31%	24%	28%	9%	62%	17%
Senior Manager	12%	53%	48%	26%	5%	55%	10%
Line Manager C.O. Range	9%	11%	11%	10%	9%	10%	3%
Software Engineer C.O. Range	3%	10%	9%	10%	6%	11%	9%
Senior Manager C.O. Range	7%	11%	11%	10%	5%	11%	6%

FIGURE 7.13 CONFIDENCE INTERVALS

Feedback	Performance	Job Satisfaction	Motivation	Attitude	Behaviour	Feelings	No Response
Line Manager	43%	91%	80%	51%	24%	77%	1%
Software Engineer	33%	68%	61%	45%	21%	76%	1%
Senior Manager	31%	82%	69%	42%	18%	66%	3%
Line Manager C.O. Range	8%	4%	6%	8%	6%	6%	1%
Software Engineer C.O. Range	7%	7%	7%	8%	6%	7%	1%
Senior Manager C.O. Range	7%	6%	7%	8%	6%	7%	3%

FIGURE 7.14 CONFIDENCE INTERVALS

Feedback	Performance	Job Satisfaction	Motivation	Attitude	Behaviour	Feelings	No Response
Line Manager	18%	41%	41%	23%	19%	69%	8%
Software Engineer	7%	32%	25%	23%	12%	65%	14%
Senior Manager	15%	52%	45%	27%	14%	63%	10%
Line Manager C.O. Range	6%	8%	8%	6%	6%	7%	4%
Software Engineer C.O. Range	4%	7%	7%	6%	5%	7%	6%
Senior Manager C.O. Range	5%	8%	8%	7%	5%	7%	7%

FIGURE 7.15 CONFIDENCE INTERVALS

Feedback	Performance	Job Satisfaction	Motivation	Attitude	Behaviour	Feelings	No Response
Line Manager	53%	94%	86%	53%	31%	89%	0%
Software Engineer	39%	61%	64%	47%	28%	83%	3%
Senior Manager	50%	86%	86%	53%	31%	78%	0%
Line Manager C.O. Range	14%	7%	10%	14%	13%	9%	0%
Software Engineer C.O. Range	14%	14%	14%	8%	13%	11%	2%
Senior Manager C.O. Range	14%	10%	10%	14%	13%	12%	0%

FIGURE 7.16 CONFIDENCE INTERVALS

Feedback	Performance	Job Satisfaction	Motivation	Attitude	Behaviour	Feelings	No Response
Line Manager	17%	36%	36%	19%	25%	69%	11%
Software Engineer	14%	31%	19%	19%	17%	67%	17%
Senior Manager	22%	50%	39%	28%	31%	72%	14%
Line Manager C.O. Range	11%	14%	14%	11%	12%	13%	8%
Software Engineer C.O. Range	10%	13%	11%	11%	11%	13%	11%
Senior Manager C.O. Range	12%	14%	14%	13%	13%	13%	10%