How do Software Developers Experience Team Performance in Lean and Agile Environments?

Fabian Fagerholm, Marko Ikonen, Petri Kettunen, Jürgen Münch
Department of Computer Science, University of Helsinki
P.O. Box 68, FI-00014 University of Helsinki, Finland
fabian.fagerholm@helsinki.fi, marko.ikonen@alumni.helsinki.fi,
petri.kettunen@cs.helsinki.fi, juergen.muench@cs.helsinki.fi

Virpi Roto
School of Arts, Design and Architecture
Aalto University
P.O. Box 31000, 00076 Aalto University, Finland
virpi.roto@aalto.fi

Pekka Abrahamsson
Faculty of Computer Science
Free University of Bozen-Bolzano
Piazza Domenicani 3, Bolzano, Italy
pekka.abrahamsson@unibz.it

ABSTRACT

Context: Companies increasingly strive to adapt to market and ecosystem changes in real time. Evaluating team performance in such changing environments presents a major challenge. **Objective:** This paper aims to understand how software developers experience performance in a highly volatile environment. This understanding could be used as a basis for guiding formation and maintenance of high-performing teams. **Method:** A qualitative multiple-case study using thematic interviews was conducted with 16 experienced practitioners in five organisations. **Results:** We found 33 major categories of performance factors, arranged as a theoretical structure that explains how the subjects experience software team performance. **Conclusions:** Based on our study, software teams are engaged in a constant cycle of interpreting their performance and aligning it with other stakeholders. Enhancing performance experiences requires integration of soft factors, such as communication, team spirit, and team identity, into the overall development process.

Categories and Subject Descriptors

D.2.9 [**Software engineering**]: Management—*productivity, programming teams*; K.6.1 [**Management of computing and information systems**]: Project and People Management—*staffing*

General Terms

Performance, Management, Human Factors, Theory

Keywords

Developer Experience, Performance, Success, Case Study, Qualitative Study, Human Factors, High-performing Teams, Agile Software Development, Lean Software Development

1. INTRODUCTION

Performance is a multi-faceted concept that is used on several levels of an organisation to mean different things [25]. The desired outcome, a successful and well-performing software product or service, is contingent on a complex combination of factors that can be found in projects, processes, organisations, teams, and individuals (e.g. [48, 44, 53, 52]). Within these categories, there are multiple characterisations of performance that are relevant in different contexts and for different purposes. Even the performance of the end result, the software itself, can be viewed in different ways; e.g. in terms of technical quality, fitness for purpose, or generated profits. Today's software development organisations aim to improve performance by being more responsive to changing market needs, e.g., by treating R&D as a continuous experimentation system [34].

When the objective is to evaluate teams, human factors are brought to the forefront. A team may be evaluated, e.g., in terms of its productivity [53], speed [5], or ability to produce novel and innovative results [39]. It may also be evaluated in terms of process control [48], or the knowledge it produces [49]. Many factors influence performance within these areas. However, since software development is largely a human-based activity, most types of outcome depend on human factors. Motivation, skill, satisfaction, values, and personality are factors to consider when forming teams, creating and designing processes and development environments, and structuring organisations and communication. The importance of such human aspects on performance in software development is well known [53, 22, 4, 31, 6]. However, there is a lack of understanding in many software development environments of how software practitioners themselves experience the pursuit of high performance, and how striving for performance could simultaneously be a meaningful and positive experience.

In this paper, we aim to cast light on how professional software developers experience performance in a Lean and Agile context. Drawing on a previous conceptualisation of Developer Experience [23], we approach the issue through a cognitive, affective, and conative lens. We view team performance from the perspective of individual software practitioners, gaining insights that may be of use in evaluating teams from an internal perspective. The study shows why it is not sufficient to consider performance only as meeting predefined objectives. It also shows how practitioners reason as they attempt to perform in their work, and what they perceive as beneficial and detrimental for those attempts. Our specific research questions are:

RQ1 How do software practitioners experience team performance in Lean and Agile environments?

RQ2 How do software practitioners reason about the relationships between perceived performance factors?

The remainder of the paper is organised as follows. In Section 2, we discuss the concept of performance in software engineering, with particular focus on human factors on the team and individual levels. In Section 3, we describe our research approach: the data collection and analysis methods used. In Section 4, we present the empirical results. We discuss the implications and limitations of our findings in Section 5. Finally, we conclude the paper in Section 6 and briefly outline possible future work.

2. THEORETICAL BACKGROUND

One of the foremost practical objectives of team performance research is the pursuit of ways to improve the work outcome of teams. It is interesting to note that teams were once considered an improvement over individual work: teams can potentially offer greater adaptability, productivity, and creativity than any single individual [55, 26, 29]. However, gaining the potential benefits of teams is not easy. For example, it is not enough to merely group skilled individuals together [30]. In this section, we briefly discuss how to define performance, and shortly review some existing research on performance factors and models of team performance.

2.1 Definition of Performance

One definition of high-performing teams is that they outperform "all reasonable expectations as well as all other similarly situated teams" [37]. While this definition proceeds to say that the performance of these teams surprises even themselves, organisations find high-performing teams highly desirable and wish to replicate their success. However, reports describing such high-performing teams are typically on an anecdotal level, based more on assumptions than on a valid causal analysis. Part of the problem may stem from the lack of a sound measure for "success" in software engineering, although it is a central dependent variable [51].

Performance is often divided into efficiency and effectiveness. Efficiency means accomplishing objectives quickly and with minimal resource usage. Effectiveness refers to accomplishing the right objectives, e.g. those that have the greatest value. However, the terms can be used differently; e.g. Salas et al. use them as follows [55]. Team performance refers to "the outcomes of the team's actions regardless of how the team may have accomplished the task". Team effectiveness considers "not only whether the team performed" (e.g. completed a task), but also "how the team interacted to achieve the team outcome" (e.g. team processes, teamwork). The distinction is important since many factors may influence the outcome, and confound the causal reasoning assumed in team performance measures. This may result in an incorrect understanding of the team [55]. In this work, we use "performance" as an umbrella term for all the meanings described above and use more specific terms as needed.

2.2 Performance Factors

Sudhakar et al. [59] list four classes of factors which influence team performance: i) technical, ii) non-technical (soft), iii) organisational, and iv) environmental. The technical factors include project-specific traits such as size, complexity, and processes, as well as product characteristics. There are numerous reported soft factors, and fully explaining them is beyond the scope of this paper. However, some examples can be mentioned.

On the individual level, cognitive factors include skill [8, 7, 59], knowledge [43], competence [31], and logical reasoning [11]. Motivation is a conative factor that has received much attention in software engineering research [24, 6]. Personal values [43], beliefs [19, 50], and personality [59, 6] have also been investigated as direct or indirect performance factors. In addition, affective factors have been examined, showing that developers do experience several emotions in their work, and that these change over time [58]. Moods can influence programming tasks such as debugging [40]. Enthusiasm [60], and emotional valence and dominance [28], can have a positive effect on performance, while frustration is a negative risk factor for performance [60].

On the group or team level, some of the reported factors include cohesion [18, 62, 35], trust [18, 1, 35], clarity of purpose and goal-setting [1], group structure and communication [14, 53, 61], knowledge sharing [31], team relationships, diversity, and leadership [59], and coordination processes [61, 41]. The organisational factors include organisational culture, climate, structure, and values [59]. Finally, the environmental factors include industry characteristics and volatility, and also factors relating to customers and competitors [59].

2.3 Team Performance Models

Many researchers have shown that team effectiveness is the result of the team's processes (e.g. [26, 29, 46]). However, it is less clear what those processes are and how they result in improved outcomes. Salas et al. [55] note that teams "do more than simply interact with tools; they require the ability to coordinate and cooperatively interact with each other to facilitate task objectives through a shared understanding [of available resources, goals and objectives, and constraints]." Research has shown that different types of teams manifest teamwork processes differently [55].

Team performance models aim to describe causal relationships between variables that result in performance outcomes or at least provide actionable advice for managing performance. Dingsøyr & Dybå discuss three teamwork models concerned with team effectiveness from an internal perspective [16]. In Salas' model [55], five components (team leadership, mutual performance monitoring, backup behaviour, adaptability, and team orientation) interact to produce performance. Three coordinating mechanisms (shared mental models, closed-loop communication, and mutual trust) are proposed as means to raise the level of performance. The Dickinson and McIntyre model [15] is similar to the Salas model. It adds feedback and coordination, and is intended for self-managed teams. The Hoegl model [32] has six facets: communication, coordination, balance of member contributions, mutual support, effort, and cohesion. The model has been shown to have a significant association with team performance (divided into effectiveness and efficiency) and team members' personal success (work satisfaction and learning).

Dingsoyr & Dybå assert that although several team performance (or effectiveness) models exist in other disciplines, there are many open questions regarding their use in software engineering [16]. The relationship between team performance and project success also remains an open question. Success includes not only meeting schedules and making profits, but also encompasses employee well-being and public impact [52]. The notion of performance must then be considered dynamic, to include the activity of defining its meaning. In this expanded definition, performance can be understood in many different ways depending on the viewpoint [52, 25], and viewpoints may conflict [52, 39, 36]. An open question is therefore how software development practitioners experience the pursuit of high performance in an uncertain environment. Could the pursuit of high performance be more than improving the work outcome of teams?

3. RESEARCH APPROACH

Due to the nature of our research questions, we chose an exploratory, embedded multiple-case study method [63]. Case studies aim at investigating contemporary phenomena in their context [54] and are suitable for research questions of an exploratory and explanatory nature [63]. Our aim is to generate, not test, theory. There are several ways in which case studies can be used to inductively build theory [21, 20]. We used an analysis strategy based on grounded theory coding methods [12].

3.1 Sample and Context

Being a multiple-case study, this study aims to understand the dynamics of performance in software development teams by viewing it from the perspective of multiple practitioners in multiple organisations. On both company and subject levels, we used maximum variation sampling [20, 27], where the focus is on finding variants on a common theme [47]. Widely varying instances are of particular interest since they allow capturing the core experiences and common dimensions of a setting or phenomenon [27]. This expands the range of applications in which the results can be used [47].

We approached five companies with offices in Finland (Table 1), varying in terms of size, industry and market. They were selected because they used a Lean and Agile approach and because they operated in volatile markets. All five agreed to participate in a research project on team performance. The companies vary in size, ranging from around 50 employees to almost 1000 employees in the Finnish location and tens of thousands worldwide. All are at least 10 years old; the oldest traces its roots back more than 100 years. One is a Fortune 500 company, two are publicly traded on Nordic stock exchanges, and two are privately owned. The companies operate in several different application domains, including telecommunications, embedded and wireless systems, data and network security, and general software and business development services. Some of them provide consultation services and product and service development to third parties, while others market their own products directly to businesses and consumers. All companies had offices in or near the Finnish capital of Helsinki at the time of the study, the large ones with several offices in different parts of Finland. All companies have adopted Lean and Agile development principles and use some variant of Lean and Agile software development in their development process. In the older and larger companies, multi-year organisational transformations have been conducted to replace earlier software development approaches with more modern ones.

Following our instructions, contact persons within each company purposively selected subjects with sufficient experience to give relevant information regarding the research topic. We sought subjects from different parts of the development organisation, to cover a wide range of perspectives. At the time of the study, the subjects worked in teams of approximately 5–15 persons. However, all of them had worked in several teams of different sizes during their career, and thus had experience with many different team conditions to draw from. In total, our sample consists of 16 practitioners, including managers on the company and department levels (3), coaches/team leaders (11), and team members (2); 13 were male and 3 female. Most practitioners were native Finns, but two subjects were not.

3.2 Data Collection

We used thematic, semi-structured interviews [3, 47] for data collection. In thematic interviews, subjects are interviewed about issues directly related to the object of study – the theme [3]. They allow quick access to a wide and deep range of practitioner expertise and are particularly useful for aspects that the interviewee is not accustomed to speak about on a daily basis, such as values, intentions, or ideals. The amount of structure in such interviews may vary. A semi-structured interview, used here, is a mix of more and less structured questions, but with flexible wording and question order. A base set of questions is always covered, but there is room for open-ended, exploratory conversation. Multiple perspectives on the same issues can thus be examined, resulting in triangulated data both within and between subjects.

Table 1: Key Facts of Case Companies.

Company	Field of Industry	Size	Type
A	Business development, consulting	Small	Local
B	Telecom networks and services	Very large	Multinational
C	Computer security	Large	Multinational
D	Embedded and wireless systems	Large	Multinational
E	Software design and development	Medium	Local

An interview guide is a list of questions to be asked or topical areas to be covered by the interviewer, possibly including their order and other instructions [47]. We designed an interview guide for discussing performance from several perspectives. The guide was designed in a chronological fashion to help recollect experiences from subjects' entire careers. It was kept flexible enough to allow constant analysis of interviews to affect the direction of subsequent interviews, supporting the grounded theory approach of constant comparison [27, 13]. The guide is shown in Appendix A.

Data collection was carried out during two months in 2011. Two researchers carried out 16 semi-structured thematic interviews of about 1.5 hours each with subjects from the five companies under study. Each interview was recorded for later reference. Notes were primarily taken by one of the researchers while the other primarily concentrated on interviewing the subject. After each interview, the researchers discussed the interviews, wrote supporting field notes, and constantly compared the interview material with previous interviews. When subjects spontaneously raised a topic or told a story, we allowed them to do so freely, while making sure that the themes in the guide were covered. Where applicable, we specifically asked follow-up questions about personal opinions, reasoning, and feelings. Interviews were carried out in English or Finnish, according to the subject's preference.

3.3 Data Analysis

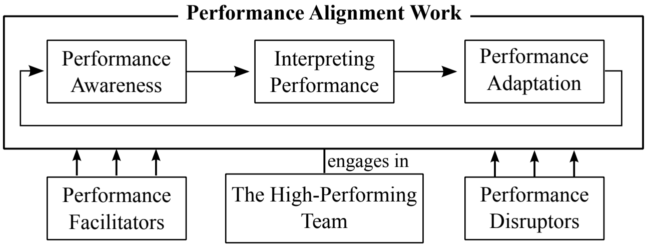
We employed coding strategies from grounded theory method [27, 12] to analyse the interview data. Grounded theory can be thought to proceed in three phases [12]: open, axial, and selective coding. During open coding, we identified categories in the data by grouping related interview fragments. Fragments that mentioned, e.g., a similar performance factor, or reasoned similarly about causes and effects, were grouped together. During open coding, related groups may also be clustered into higher-order categories. In axial coding, we related categories to each other, creating a refined category scheme with links between the categories.

To enable multiple researchers to collaboratively perform open and axial coding, we used the Affinity wall method. The method originates in quality management research [38] but has been transferred into system design as a tool for consolidating large amounts of seemingly disparate information into a coherent picture [10, 9]. The Affinity wall method ensures that several researchers consider and discuss each and every piece of data, bringing researcher triangulation into a central position in the research process. The method involves multiple participants iteratively categorising pieces of information written on paper notes onto a flat surface, usually a wall. In our case, the notes were self-contained pieces of interview data. The result is an Affinity diagram, a hierarchical diagram which structures field data (notes) into conceptually similar groups, which are then further organised into higher-level categories. The diagram is often referred to simply as an Affinity wall.

In the final phase, we used selective coding. Selective coding develops a core category, propositions, or a hypothesis. Here, analysis does not seek to summarise material without losing detail as in the Affinity wall method, nor to expand to generate new perspectives, but aims to proceed quickly and selectively towards a coherent, integrated theory [56, 27]. In our case, the overall understanding in the data pointed towards a core category that described a sense-making and negotiation process.

4. RESULTS

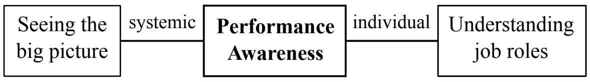
The core category *Performance Alignment Work* was constructed to summarise the entire data set. This category refers to the continuous process that all subjects were engaged in to negotiate the meaning of performance in different situations, interpret their current performance, and adapt it to changing circumstances. In this section, we introduce the categories around the core category and relationships among them that arose during the analysis. We illustrate the results using in-text diagrams. We include interview quotes as examples of the data behind the categories. Each category is emphasised in the text. In Table 2, we also show the 33 categories that emerged from the Affinity wall analysis, to facilitate traceability into the interview material.



4.1 The Meaning of Performance

Two top-level categories form two axes that summarise the differing views among subjects regarding the meaning of performance. *Performance awareness* describes the level of self- or other-orientation in subjects' perception of performance. This axis forms a continuum ranging from the individual (*Understanding job roles*) to the team, unit, organisation, and market (*Seeing the big picture*).

> "[The way I was trained], the project manager had to know everything about everything. He had to think about everything. I realised that in Agile, the team has the power to decide, and is trusted." (*Coach, Company E*)

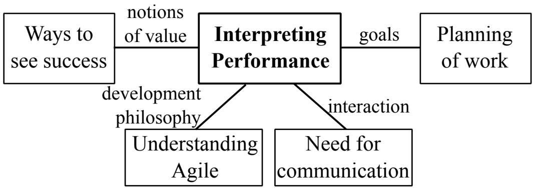


Interpreting performance is another axis that describes the desirable level of performance: meeting or exceeding predefined objectives – or transcending them altogether by participating in their definition and assessment. These interpretations stem from different *Ways to see success*, different understandings of the development philosophy (*Understanding Agile*), different views on the desirability of being involved in social interaction (*Need for communication*) and different notions of how goals should be set and pursued (*Planning of work*).

Table 2: Categories from the Affinity wall. The number of supporting second- and third-level sub-categories, and the companies from which the supporting interview data originated, are shown. The table is sorted by number of third-level subcategories, followed by the number of source companies.

Top-level category	Number of sub-categories 2nd level	3rd level	Companies A	B	C	D	E
Need for communication	6	36	×	×	×	×	×
Team spirit	8	36	×	×	×		×
Improving the process	7	33	×	×	×	×	×
Re-organisation	7	31		×	×	×	×
Team setup	8	30	×	×	×	×	×
Tools	8	30		×	×	×	×
Decision power	6	29		×	×	×	×
Facilitating communication	3	29		×	×	×	×
Organisational learning	5	27	×	×	×	×	×
Organisational support	8	26	×	×	×	×	×
Planning of work	6	26		×	×	×	×
Ways to see success	6	25	×	×	×	×	×
Reward	6	24	×	×	×	×	×
Atmosphere	5	24	×	×	×		×
Time investment	6	23	×	×	×	×	
Seeing the big picture	5	23		×	×	×	×
Collaboration and cooperation	3	23		×	×	×	×
Personal development	4	21		×	×	×	×
Understanding job roles	4	21		×	×	×	×
Distributed work	5	21	×	×	×	×	
Team identity	5	20	×	×	×	×	×
Goal setting	6	20		×	×	×	×
Understanding Agile	4	19	×	×	×		×
Adapting to change	4	18	×	×	×	×	×
Pride	4	17		×	×		×
Control of my own work	4	16		×	×	×	×
Social skills	5	16		×	×	×	×
Intrinsic motivation to perform	4	15		×	×	×	×
Prioritisation	4	14		×	×	×	×
Learning from failures	4	14	×	×	×	×	×
Testing	4	14		×	×	×	×
Open office	3	11		×	×		×
Getting buy-in	3	10			×	×	×
Total	170	742					

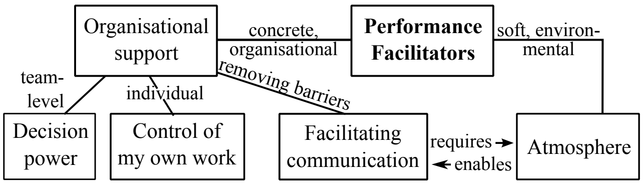
> "Good performance is such that it fulfils expectations and the expectations come from some kind of conception about the end customer." (*Manager, Company D*)



4.2 Factors Impacting Performance

Subjects reported on factors that they perceived to facilitate and disrupt performance. In addition, they reported on factors that they saw as having both a positive and negative effect. Two types of *Performance Facilitators* were reported: factors that the organisation could concretely influence (*Organisational support*), and soft environmental factors (*Atmosphere*). *Decision power* placed within teams, and individual autonomy in how tasks are carried out (*Control of my own work*) were seen as two positive factors that the organisation should implement. Removing barriers and *Facilitating communication* was also seen as the responsibility of the organisation. This factor was linked to *Atmosphere*, since open communication requires a supportive environment and vice versa. Creating a good atmosphere by facilitating communication was seen as important. Practitioners indicated both that the presence of these factors were beneficial for performance, but also that their absence was detrimental.

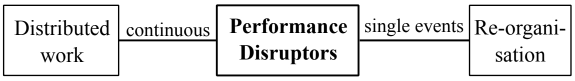
> "Last summer our team worked really well, everyone knew what everyone else was doing. We didn't need any formal meetings, communication was natural and direct. Whenever someone from the outside asked us what we were doing, we were able to give a direct answer about our current status and give predictions on when different things would be ready. It was fun to come to work because everything just worked without any extra challenge or effort. [Then] a colleague and I went to work abroad for half a year. When we returned, something had happened to the team. It didn't work any more. [People] didn't talk to each other in the same way as before." (*Team leader, Company B*)



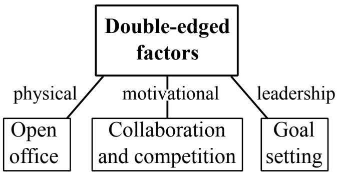
Two strong subcategories emerged as *Performance Disruptors*. *Distributed work* was seen as having a continuous negative influence on performance, but one which is manageable through increased emphasis on communication. *Re-organisations* were seen as events with a major negative performance impact that requires both time and effort to recover from.

> "When the team changes, all forecasts on team performance like velocity estimates become invalid. I am trying to figure out all the time how much work we can do when the skill set and motivation changes in the team. The [layoff negotiations] have been kind of a trauma in the team. We had to find ways to motivate people to take the work that was left over from the team which was shut down. I think this was the darkest period of the team and my career in this company. The team talks about the dark period from time to time to remember that we changed. Not survived, but evolved." (*Coach, Company C*)

These were not the only factors that could be construed as negative impact factors, but they were the only ones that were clearly indicated as such in the material. Other negative factors were not as clearly distinguished and their connections to other concepts caused them to become part of other categories.



Three *Double-edged factors* were described. The potential of an *Open office* to facilitate communication was seen as positive. However, the amount of communication could become disruptive, and subjects reported on the need for norms and behavioural signals to avoid the drawbacks. The category *Goal setting* refers to the balance between goals being set outside and within the team. Some subjects were firmly against goals being set outside the team because the team had the information needed to set them realistically. Others felt that the team did not always have the expertise or information required to set goals, and that outside guidance in these cases could be beneficial. *Collaboration and competition* included comments regarding competition between teams. While some reported temporary higher performance as a result, cross-team collaboration was seen as more motivational.



Two interrelated categories relate to using automation to facilitate performance. The category *Testing* reflects the primary means by which subjects approached technical quality. *Tools* were frequently mentioned in relation to software development, testing, and communication. Subjects reported that they deeply integrate tools into their development process, to the extent that their process-related discourse contains terminology and jargon borrowed from the tools themselves.

> "We have information radiators that show the condition of the code in the version control system. Before we started using git and other related tools, we didn't have very strict control over our code. During the last months, we have made stricter rules. The code in the master branch has to be in very good shape, so that merges and other version control operations work properly." (*Team leader, Company B*)

However, the tools do not necessarily need to be sophisticated; a pragmatic approach was often favoured by the subjects.

> "We used an electronic tool for planning, but it didn't really work. Then we just started putting notes on the wall for everyone to see. [...] Technology is seldom the solution, but it can add efficiency." (*Team leader, Company C*)

4.3 Adapting Performance

Three categories emerged that described how subjects adapted their performance when conditions changed. First, the whole notion of adaptation emerged as an attitude towards *Adapting to change*. Practitioners differed in whether they viewed adaptation as necessary or desirable at all, depending on whether they perceived the source of change as external to the organisation or not. If they did, their attitude was more favourable towards adaptation.

> "When the world is changing, there is no such thing as the old, familiar, and safe. You need to change the way of working." (*Coach, Company D*)

Concrete ways of adaptation centred around *Improving the process*, but the notion of what the process is and how it should be improved differed. One view was that processes are beneficial because they formalise and capture procedures that can be reliably replicated across large parts of the organisation. They were seen as a *Time investment* that must be pragmatically balanced against the gained benefits. Another view was that processes formalise and improve decision-making, helping to avoid overwork and biased decisions through *Prioritisation* procedures.

> "[In the past], there was a constant fight [among project managers on] whose task gets priority. The project managers fighting over team resources used to say that it is impossible to set priorities for new development tasks and bug fixing tasks. We can laugh at it now that we have clear priority order on the Kanban board." (*Coach, Company B*)

A third view was that processes need to be more fine-grained and separate for different parts of the development cycle. This reflected a preference for teams owning their own processes, selecting and developing them for their own particular needs. Subjects that expressed this view also viewed processes as ephemeral: they felt that

Lean and Agile values and the spirit of a continuous search for better performance were more important than following methods to the letter.

> "Steering mechanisms are completely different in the starting end of an innovation funnel than in the end. In the beginning of the innovation funnel, the impact of a single person is large. A firm process at the beginning of the innovation funnel will kill innovation. At the end of an innovation funnel, the cost and impact of choices grow, and therefore decision-making and steering mechanisms are different at different stages." (*Management team leader, Company A*)

Another concern for the subjects was how to propagate adaptive actions across the organisation (*Organisational learning*).

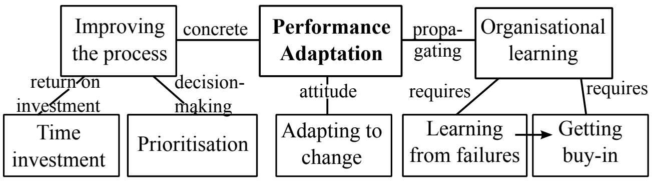
> "A good organisation does not change things just for the sake of change. [It] really learns from its mistakes, and does not make them again, [but] the larger an organisation, the harder it is to get it to work well." (*Team leader, Company B*)

Learning from failures was seen as occurring first locally in teams, and the challenge was then to convince other teams and the rest of the organisation to adopt the solution (*Getting buy-in*).

> "When there is a problem, it's everyone's problem. We do not look for a person to blame, but we try to find out why we did not see the problem coming in advance." (*Manager, Company B*)

> "I think the Scrum master needs to let the team fail – once. After the failure, we should discuss together how to avoid such a failure in the future. In practice, it is hard to let the team fail and learn from the failure. Nobody wants to take the [temporary decrease in performance]." (*Coach, Company C*)

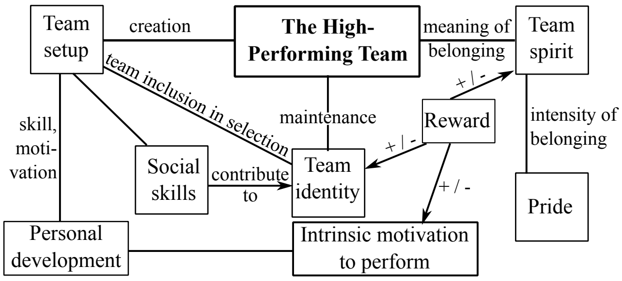
> "The best way to improve the work of the surrounding organisation is by example. Showing how we do Agile work is more effective than lecturing or forcing. When we increase the visibility of our way of working, little by little elements of it sneak into other teams as well." (*Developer, Company E*)



4.4 The High-Performing Software Team

Views on high-performing software teams formed a distinct category in the interview material. During our analysis, we grouped data fragments and categories related to this theme into a structure that explains how our subjects reasoned about high-performing teams. Descriptions of these teams often depicted them as self-directed and resourceful:

> "[Based on] my experience, good teams usually have a do-it spirit like a small company. They have established a flow through which people commit to their work and see it as interesting, important and the right thing to do. They put in more than just office hours. [These teams] were formed in an unusual way. There was no line organisation, really not much of an organisation at all. The organisation had a largely fractal structure that changed all the time, like a kaleidoscope. The core is an individual or a team which gets an idea that is larger than life. They combine competencies inside and outside the company and produce a product or service. They find investors for their project, customers, and resources to implement their idea." (*Management team leader, Company A*)



We found some reasoning on the teamwork processes of high-performing software teams. A first category concerned the creation of high-performing teams (*Team setup*). Subjects expressed that *Social skills* should be a selection factor for such teams.

> "There are some people who do not fit into a Scrum team and its active way of working. I think that if we are looking for very efficient teams, then their members will be in one end of a scale of social skills. Multitalented people are usually socially skilled. [...] People who do not have the social skills to work in a Scrum team could work in some other kind of team that does not develop itself or have these scary social practices." (*Coach, Company C*)

Social skills contribute to the formation and maintenance of a powerful *Team identity*. Subjects expressed that the team should be included in the selection process to preserve the identity. They expressed that this applies also in the case of adding a new member into the team. Subjects indicated that on the individual level, *Intrinsic motivation to perform* spurs a desire for *Personal development*.

> "[I'm motivated because] I'm solving my problems for the next two years. [That] motivates to start learning a new area." (*Coach, Company C, discussing setting up a new continuous integration system*)

> "We are a company that provides you a chance to learn and develop [yourself] better than anywhere else. [...] In the new organisation, everyone knows that it is important to develop oneself in order to be competitive." (*Manager, Company B*)

Thus subjects were motivated to join high-performing teams to both express and develop their own skills, and to gain opportunities for high performance. According to some subjects, existing teams will respond to that kind of motivation and welcome individuals who are prepared to work hard for exceptional results.

> "The team I worked in previously decided to take as our fundamental mode of work to be the best team in the company. When a new guy joined our team, who had been in the company for several years, he said he had never had to work as hard as in our team." (*Team leader, Company C*)

Some subjects expressed strong reluctance against letting individuals without these traits join their teams, reflecting a maintenance strategy of the *Team identity*.

> "We have chosen people who have an excellent track record, and not tolerated anyone who is not excellent. It might be that it would be better for [a person] who can't work in an Agile way to work elsewhere." (*Team leader, Company C*)

Other subjects took a more inclusive view, reflecting a different kind of identity:

> "I feel that having seen the world a little bit, nobody is really that much more [competent] than anyone else. It's always a case of 'personal chemistry'. It is sometimes forgotten that you need someone to facilitate inclusion into the team. Someone who says 'hey, come and join us'." (*Coach, Company E*)

While becoming part of a high-performing team appears to be associated with some challenges, belonging to one can be a meaningful experience. Subjects expressed that such teams have a special *Team spirit*: a feeling of belonging to a group of like-minded individuals with a high degree of respect for each others' talents.

> "More than a company, this is a coalition of intelligent people who think alike." (*Management team leader, Company A*)

> "A good way of working has more to do with the the work spirit than with the methods themselves. If you have a good work spirit, then you are automatically considerate towards others, both in what the team does and how it does it." (*Team leader, Company B*)

When this feeling of belonging was intense, subjects expressed a feeling of *Pride*.

> "I feel good when I know I've done good quality work. If I cannot be proud of the outcome of my work, I get frustrated. The possibility of having an influence on my own work drives me." (*Developer, Company E*)

Finally, subjects expressed how *Reward* could influence the dynamics of the high-performing team. The process of obtaining a reward was seen as more important than the reward itself. It should be linked to an actual episode of good performance, preferably one that the team can influence.

> "Promising rewards like money, free trips, or a car to use, in return for meeting a deadline is the wrong place to start. If the team itself wants some kind of rewards in the form of doing things together, then that is something to encourage." (*Team leader, Company B*)

A badly motivated or timed reward, given without consideration of how the team itself has perceived its performance, may result in dissonance and emotional rejection of the reward, impacting *Intrinsic motivation to perform*, *Team identity*, and *Team spirit* negatively. Also, the reward should be valued by the team – a social experience could have a better effect than individual monetary rewards.

> "It was almost impossible to reach the given target in the given schedule, but we worked night and day because we had a common dream about enjoying the sunset in Mexico with the whole team as a reward. And we did it." (*Manager, Company B, discussing a successful project experience*)

5. DISCUSSION AND LIMITATIONS

Our results indicate that the practitioners in our sample experience performance as a continuous process of negotiation within their teams and with external stakeholders. They are aware of performance aspects on multiple levels of the organisation. They perceive high-performing teams in terms of group processes that link skilled and motivated developers to a powerful team identity. In this section, we discuss our research questions in light of the results, and consider the potential wider implications of the findings for practice.

5.1 Addressing the Research Questions

Considering RQ1, our results point to the close connection in practitioners' experience between performance and success. They support the claim that practitioners experience both as multifaceted, socially negotiated, changing over time, and sometimes as conflicted between different stakeholders (compare to "success" in Ralph et al. [52]). In this study, Performance Alignment Work is the activity by which practitioners deal with the fluidity of the performance concept. It contributes to the body of knowledge by describing a particular type of teamwork that specifically addresses performance goals and the process by which software teams attempt to reach them (see e.g. Salas et al. [55] for a discussion of teamwork).

The reported experiences have similarities to earlier findings on team performance (e.g. [17, 55, 15, 32]): as would be expected, many of the concerns expressed by our subjects revolve around communication, coordination, and group dynamics. A detailed comparison would be interesting, but is not within the scope of this paper. We instead make a comparison to the most similar study we have found. In Dingsøyr & Lindsjorn [17], study subjects take a "forcing" approach to code empirical data into the Salas model [55]. Our results are similar despite the lack of such an anchoring in our study design. Interestingly, our results are more similar in type to the original Salas model [55]: we describe a process rather than a list of impact factors. The question is what type of result is most relevant for a particular need, and future research should carefully choose a research design that produces the desired type of result.

RQ2 concerned how practitioners reason about the relationships between perceived performance factors. Two main findings arose from our analysis. The perceived factors concern on one hand the components of social negotiation of performance (Performance Alignment Work, Sections 4.1–4.3) and on the other hand an understanding of why some teams are high-performing (Section 4.4). The components of Performance Alignment Work show that our subjects reasoned about performance on all levels from their individual and team performance to the performance of their organisation in the marketplace and in terms of customer satisfaction. This indicates that practitioners have local or personal theories and beliefs regarding how their work influences team and organisational performance. Activities that raise these theories and beliefs into awareness and help align them could help improve actual performance in teams and organisations. For example, performance measurement programs could be conducted as participative design programs in which teams were deeply involved. A challenge then is how to balance the concerns of different stakeholders when they conflict, a question which ultimately requires consideration of ethics and values.

Based on our findings, a high-performing team is one that is exceptionally good at Performance Alignment Work. Not only can it continuously organise itself internally to optimise performance, it also engages with other teams, other parts of its organisation, and with stakeholders outside the organisation. It both elicits their performance needs and preferences, but also influences their performance expectations and alignment in a manner that is favourable to its goals. Our description of the high-performing team contributes to theory in two ways. First, it provides a proposition that is testable in specific cases: that high-performing teams are considered high-performing because they influence the criteria by which they are judged. Second, it shows that there are important affective aspects, and conative aspects beyond motivation, that should be considered in research on team performance. The affective aspects were present in practitioners' notions of a team identity, team spirit, pride, and in their reasoning of rewards as tools for increasing performance. Identification with a team has been established as a determinant of affective outcomes such as job satisfaction [45]. We note that an interesting question is how to characterise team identity, its formation, and its relationship to company identity, corporate culture, and values. Weak indications in our data cause us to suspect that there were important differences in corporate culture between the companies in our sample. For example, fewer categories were supported in the small Company A (see Table 2).

Research results on motivation have so far been inconclusive regarding how and by what software engineers are motivated, and what the benefits of motivating them are [6, 24]. The conative aspects in the high-performing team description imply that (intrinsic) motivation on the individual level does not directly lead to performance on the team level. There are also volitional processes involved as teams go through the process of accepting a new member. Aligning affective and conative aspects within the team could be another means to reach better performance. However, due to personal characteristics, e.g., personality and values, it may be more pragmatic to strive for such alignment when constructing teams rather than trying to change its members afterwards. Further research on these topics is called for.

5.2 Threats to Validity

The theoretical contributions of this paper are developed in, and describe, the local situation in the participating companies. The theory itself has been subject to accuracy-checking procedures – triangulation on subject, company, and researcher levels – and can be judged in terms of how well it reflects the reality of the subjects and the phenomena under study [2]. We gathered feedback on the emerged categories from one senior representative from each company who was not an interview subject. The representatives indicated that all categories were important, and they found it difficult and essentially meaningless to give the categories a forced order of prioritisation: they saw the performance phenomenon as holistic rather than in terms of individual pieces. Our interpretation is that the study reflects subjects' reality well. With certain limitations, qualitative findings may be generalised to some broader theory through replication [63].

Interviews have inherent threats to validity that relate to the ability and willingness of subjects to report on desired topics [63], and to the possibilities of generalising interview-based findings [42]. In planning the interviews, we chose interview questions that allow various kinds of factors to emerge: the questions do not favour one aspect of high performance over another. Bias was alleviated in data collection by having two participating researchers, and in analysis by having four researchers discuss and agree on the emerging results. The analysis methods ensure that an individual piece of information cannot dominate the overall result, since it must be matched with other pieces of information to form a meaningful higher-order structure. The coding methods used in this study are well known, and thoroughly documented in the literature [38, 10, 9, 13, 47, 56, 27].

A certain measure of convenience sampling is almost always present in practical studies [47]. There is an apparent bias in the sample towards coaches and team leaders. However, the sample represents persons who are highly regarded in their respective organisations, who have had exposure to many different teams and projects, and who have experienced both high and low performance in different situations. Results could differ with practitioners having considerably less or more experience.

Since the study was conducted in a Finnish setting, cultural bias is an important consideration. In Hofstede's cultural dimensions [33], Finland belongs to a cluster of primarily western countries, but with important unique characteristics: low power distance, high individualism and femininity (preferring quality of life over being the best), medium high uncertainty avoidance, and a short-term orientation. Cultures differently positioned on these scales could be expected to emphasise different aspects than in our material. However, the categories show a fairly even representation of perceptions across Hofstede's dimensions. Both individualistic (e.g. Personal development and Intrinsic motivation to perform) and collectivist points of view were represented (e.g. Team identity and Team spirit). It is also important to remember that corporate culture may partially override national culture at work, particularly in multinational corporations [57]. Nevertheless, culture should be considered when attempting to generalise the findings.

6. CONCLUSIONS

In this paper, we reported on a study that explores how practitioners experience and reason about team performance in a changing environment. We conducted a multiple-case study in which we interviewed practitioners from five companies that use Lean and Agile approaches and operate in volatile markets.

Through the study, we showed why it is not sufficient to consider performance only as meeting predefined objectives: objectives themselves change and are subject to an interpretive dialogue in which software teams can be an influential stakeholder. Practitioners understand performance on many levels, ranging from individuals and teams to organisations, markets, and customers. They hold complex local theories and beliefs regarding performance and the mechanisms that result in high or low performance, as interpreted by different stakeholders. Our expectation is that a better understanding of the experience of performance is an important component in improving work conditions while also improving actual performance.

We envision three future directions of research. First, we hypothesise that the experience of performance arises from basic social psychology, e.g. beliefs, norms, and values. A better understanding of how to apply these in software engineering research could help explain specific understandings of performance. Second, the link between software development paradigms and the experience of developers is open to enquiry. How, for instance, do practitioners alter their behaviour as a result of interpreting the principles of Lean and Agile software development? Third, what are the relationships between different stakeholders' evaluations of performance? For example, what aspects of good performance do developers and customers agree or disagree on? Accounts of such agreement and disagreement could help software development organisations to improve the experience of both developers and customers.

7. REFERENCES

[1] S. Acuña, M. Gómez, and N. Juristo. Towards understanding the relationship between team climate and software quality – a quasi-experimental study. *Empirical Software Engineering*, 13(4):401–434, 2008.

[2] D. L. Altheide and J. M. Johnson. Criteria for assessing interpretive validity in qualitative research. In N. K. Denzin and Y. S. Lincoln, editors, *Handbook of qualitative research*, pages 485–499. Sage Publications, Inc, Thousand Oaks, CA, 1994.

[3] P. Åstedt-Kurki and R.-L. Heikkinen. Two approaches to the study of experiences of health and old age: the thematic interview and the narrative method. *Journal of Advanced Nursing*, 20(3):418–421, 1994.

[4] N. Baddoo, T. Hall, and D. Jagielska. Software developer motivation in a high maturity company: A case study. *Software Process Improvement and Practice*, 11(3):219–228, 2006.

[5] S. Bannerman and A. Martin. A multiple comparative study of test-with development product changes and their effects on team speed and product quality. *Empirical Software Engineering*, 16(2):177–210, 2011.

[6] S. Beecham, N. Baddoo, T. Hall, H. Robinson, and H. Sharp. Motivation in Software Engineering: A systematic literature review. *Information and Software Technology*, 50(9–10):860–878, 2008.

[7] G. Bergersen and J.-E. Gustafsson. Programming Skill, Knowledge, and Working Memory Among Professional Software Developers from an Investment Theory Perspective. *Journal of Individual Differences*, 32(4):201–209, 2011.

[8] G. Bergersen, J. Hannay, D. Sjoberg, T. Dyba, and A. Karahasanovic. Inferring Skill from Tests of Programming Performance: Combining Time and Quality. In *International Symposium on Empirical Software Engineering and Measurement*, pages 305–314, 2011.

[9] H. Beyer and K. Holtzblatt. Contextual design. *Interactions*, 6:32–42, January 1999.

[10] H. Beyer and K. Holztblatt. *Contextual Design*. Morgan Kaufmann Publishers, 1997.

[11] G. Calikli and A. Bener. Empirical Analyses of the Factors Affecting Confirmation Bias and the Effects of Confirmation Bias on Software Developer/Tester Performance. In *Proceedings of the 6th International Conference on Predictive Models in Software Engineering*, pages 10:1–10:11, New York, NY, USA, 2010. ACM.

[12] J. Corbin and A. Strauss. *Basics of qualitative research: Techniques and procedures for developing grounded theory*. SAGE Publications, Inc., Thousand Oaks, CA, 3 edition, 2007.

[13] J. Creswell. *Research design: Qualitative, quantitative, and mixed methods approaces*. SAGE Publications, Inc., 3 edition, 2009.

[14] B. Curtis, H. Krasner, and N. Iscoe. A field study of the software design process for large systems. *Communications of the ACM*, 31(11):1268–1287, 1988.

[15] T. Dickinson and R. McIntyre. *A conceptual framework of teamwork measurement*. Psychology Press, 1997.

[16] T. Dingsøyr and T. Dyba. Team effectiveness in software development: Human and cooperative aspects in team effectiveness models and priorities for future studies. In *5th International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 27–29, 2012.

[17] T. Dingsøyr and Y. Lindsjørn. Team Performance in Agile Development Teams: Findings from 18 Focus Groups. In H. Baumeister and B. Weber, editors, *Agile Processes in Software Engineering and Extreme Programming*, volume 149 of *Lecture Notes in Business Information Processing*, pages 46–60. Springer Berlin Heidelberg, 2013.

[18] S. Dorairaj, J. Noble, and P. Malik. Understanding Lack of Trust in Distributed Agile Teams: A grounded theory study. In *16th International Conference on Evaluation and Assessment in Software Engineering*, pages 81 – 90, 2012.

[19] N. Douglas and T. Wykowski. *From Belief to Knowledge: Achieving and Sustaining an Adaptive Culture in Organizations*. CRC Press, USA, 2010.

[20] K. M. Eisenhardt. Building Theories from Case Study Research. *The Academy of Management Review*, 14(4):pp. 532–550, 1989.

[21] K. M. Eisenhardt and M. E. Graebner. Theory Building From Cases: Opportunities And Challenges. *Academy of Management Journal*, 50(1):25–32, 2007.

[22] A. Endres and D. Rombach. *A Handbook of Software and Systems Engineering. Empirical Observations, Laws and Theories*. The Fraunhofer IESE Series on Software Engineering. Addison Wesley, 2003.

[23] F. Fagerholm and J. Munch. Developer Experience: Concept and Definition. In *International Conference on Software and System Process*, pages 73–77, 2012.

[24] A. C. C. Franca, T. B. Gouveia, P. C. F. Santos, C. A. Santana, and F. Q. B. da Silva. Motivation in software engineering: A systematic review update. In *15th Annual Conference on Evaluation and Assessment in Software Engineering*, pages 154–163, 4 2011.

[25] M. Freeman and P. Beale. Measuring project success. *Project Management Journal*, 23(1):8–17, 1992.

[26] D. L. Gladstein. Groups in Context: A Model of Task Group Effectiveness. *Administrative Science Quarterly*, 29(4):499–517, 1984.

[27] B. Glaser and A. Strauss. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Transaction, Chicago, 1967.

[28] D. Graziotin, X. Wang, and P. Abrahamsson. Are happy developers more productive? The correlation of affective states of software developers and their self-assessed productivity. *Lecture Notes in Computer Science*, 7983 LNCS:50–64, 2013.

[29] J. Hackman. The design of work teams. In J. Lorsch, editor, *Handbook of organizational behavior*, pages 315–324. Prentice Hall, Englewood Cliffs, NJ, 1987.

[30] J. Hackman. Why teams don't work. In R. Tindale, L. Heath, and J. Edwards, editors, *Theory and research on small groups*, pages 245–267. Plenum, New York, 1998.

[31] T. Hall, D. Jagielska, and N. Baddoo. Motivating developer performance to improve project outcomes in a high maturity organization. *Software Quality Journal*, 15(4):365–381, 2007.

[32] M. Hoegl and H. G. Gemuenden. Teamwork Quality and the Success of Innovative Projects: A Theoretical Concept and Empirical Evidence. *Organization Science*, 12(4):435–449, 2001.

[33] G. Hofstede. *Culture's Consequences: International Differences in Work-Related Values*. SAGE Publications, California, abridged edition, 1980.

[34] H. Holmström Olsson, H. Alahyari, and J. Bosch. Climbing the "Stairway to Heaven" – A Mulitiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software. In *38th EUROMICRO Conference on Software Engineering and Advanced Applications*, pages 392–399, 2012.

[35] R. B. Hyman. Creative chaos in high performance teams: An experience report. *Communications of the ACM*, 36(10):57–60, 1993.

[36] M. Ikonen and P. Abrahamsson. Operationalizing the Concept of Success in Software Engineering Projects. *International Journal of Innovation in the Digital Economy*, 2(3):11–37, 2011.

[37] J. R. Katzenbach and D. K. Smith. *The Wisdom of Teams: Creating the High-Performance Organization*. McKinsey & Company, Inc., New York, NY, 1993.

[38] J. Kawakita. *The Original KJ Method*. Kawakita Research Institute, Tokyo, 1982.

[39] P. Kettunen. Bringing Total Quality in to Software Teams: A Frame for Higher Performance. In B. Fitzgerald, K. Conboy, K. Power, R. Valerdi, L. Morgan, and K.-J. Stol, editors, *Lean Enterprise Software and Systems*, volume 167 of *Lecture*

Notes in Business Information Processing, pages 48–64. Springer Berlin Heidelberg, 2013.

[40] I. A. Khan, W.-P. Brinkman, and R. M. Hierons. Do moods affect programmers' debug performance? *Cognition, Technology & Work*, 13(4):245–258, 2011.

[41] R. E. Kraut and L. A. Streeter. Coordination in software development. *Communications of the ACM*, 38:69–81, March 1995.

[42] S. Kvale. *Doing Interviews*. SAGE Publications Ltd, Thousand Oaks, CA, 2007.

[43] T.-P. Liang, C.-C. Liu, T.-M. Lin, and B. Lin. Effect of team diversity on software project performance. *Industrial Management and Data Systems*, 107(5):636–653, 2007.

[44] K. R. Linberg. Software developer perceptions about software project failure: a case study. *Journal of Systems and Software*, 49(2–3):177–192, 1999.

[45] A. Marks and C. Lockyer. Debugging the system: the impact of dispersion on the identity of software team members. *The International Journal of Human Resource Management*, 16(2):219–237, 2005.

[46] J. E. McGrath. Toward a "theory of method" for research on organizations. *New perspectives in organization research*, 533:533–547, 1964.

[47] S. Merriam. *Qualitative research: a guide to design and implementation*. Jossey-Bass, 2 edition, 2009.

[48] S. R. Nidumolu. A comparison of the structural contingency and risk-based perspectives on coordination in software-development projects. *Journal of Management Information Systems*, 13(2):77–113, 1996.

[49] I. Nonaka, R. Toyama, and A. Nagata. A firm as a knowledge-creating entity: a new perspective on the theory of the firm. *Industrial and corporate change*, 9(1):1–20, 2000.

[50] C. Passos, A. Braun, D. Cruzes, and M. Mendonca. Analyzing the Impact of Beliefs in Software Project Practices. In *International Symposium on Empirical Software Engineering and Measurement*, pages 444–452, 2011.

[51] P. Ralph, P. Johnson, and H. Jordan. Report on the First SEMAT Workshop on General Theory of Software Engineering. *SIGSOFT Software Engineering Notes*, 38(2):26–28, 2013.

[52] P. Ralph and P. Kelly. The Dimensions of Software Engineering Success. In *Proceedings of the 36th International Conference on Software Engineering*, New York, NY, USA, 2014. ACM. (To appear.).

[53] R. Rasch and H. Tosi. Factors affecting software developers' performance: An integrated approach. *MIS Quarterly: Management Information Systems*, 16(3):395–409, 1992.

[54] P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164, 2009.

[55] E. Salas, D. E. Sims, and C. S. Burke. Is there a "Big Five" in Teamwork? *Small Group Research*, 36(5):555–599, 2005.

[56] J. Saldaña. *The Coding Manual for Qualitative Researchers*. SAGE Publications, Ltd., 2009.

[57] S. C. Schneider. National vs. Corporate Culture: Implications for Human Resource Management. *Human Resource Management*, 27(2):231–231, 1988.

[58] T. Shaw. The Emotions of Systems Developers: An Empirical Study of Affective Events Theory. In *Proceedings of the 2004 SIGMIS Conference on Computer Personnel Research: Careers, Culture, and Ethics in a Networked Environment*, SIGMIS CPR '04, pages 124–126, New York, NY, USA, 2004. ACM.

[59] G. P. Sudhakar, A. Farooq, and S. Patnaik. Soft factors affecting the performance of software development teams. *Team Performance Management*, 17(3):187–205, 2011.

[60] M. Wrobel. Emotions in the software development process. pages 518–523, Gdansk, Sopot, 2013.

[61] H.-D. Yang, H.-R. Kang, and R. M. Mason. An exploratory study on meta skills in software development teams: Antecedent cooperation skills and personality for shared mental models. *European Journal of Information Systems*, 17(1):47–61, 2008.

[62] H.-L. Yang and J.-H. Tang. Team structure and team performance in IS development: A social network perspective. *Information and Management*, 41(3):335–349, 2004.

[63] R. Yin. *Case study research: design and methods*. SAGE Publications, Inc., 4 edition, 2009.

APPENDIX

A. INTERVIEW GUIDE

Table A1 shows the guide used for the thematic interviews in this study. The guiding questions are examples of questions that were used to transition into the thematic areas. The application of the guide was informed by constant comparison and purposive sampling (see main text, Section 3).

Table A1: Guide for thematic interviews.

Theme	Guiding questions	Purpose
Personal background	When did you start working here? In what role? What kind of role and job do you have now in this company?	Help subject recall early stages of career and state of company at that time. Elicit comparison with current role.
Team	Are you part of some team, or several? How long has this team existed? Have you been involved in it from the beginning? When did you become involved? How has the team composition changed along the way?	Understand subject's previous and current involvement in a team/teams.
Work environment / Organisation	How has this company changed during your time here? Have you encountered Agile and Lean in your work? How has this (Agile/Lean/other principle) changed the working habits of your team?	Elicit discussion about subject's views on the organisation and working environment. In particular, Agile and Lean topics inform us how the organisation functions.
Experience examples	Give examples of successful (unsuccessful) work experiences. On what scale did you succeed/fail?	Help subject to recall concrete and meaningful experiences of performance (high or low) and success (failure).
Team performance	In what way and why do you think you succeeded (failed)? What would be needed for a team to always do as well as or better than in your example? What do you think was the reason for the failures? Did the failure somehow benefit the team or the organisation?	Situate the concrete examples in a team context and elicit discussion about causes and effects, and about potential organisational learning.
Quality	What is a good (bad) team? What is a good (bad) organisation? What is a good (bad) result or product? What are your thoughts about good work, good ways of working, and well-being at work?	Elicit discussion about quality and meaning of teams, organisations, products, and work.