The appearance and effects of the planning fallacy bias in agile IT projects

Haenen, I. (Isabelle)

The appearance and effects of the planning fallacy bias in agile IT projects

—

De manifestatie en effecten van de planning fallacy bias in agile IT-projecten

Program:	Open University of the Netherlands, Faculty of Beta Sciences Master Business Process Management & IT
Courses:	IM0602 Preparation to Graduation BPMIT IM9806 Graduation Assignment Business Process Management and IT
Student:	Isabelle Haenen
Identity number:	
Date:	28-01-2021
Supervisor:	Nick Benschop
Co-reader:	Vanessa Dirksen
Version:	1.0
Status:	Final

Abstract

Due to the complexity of IT projects the decision making is often plagued by a variety of cognitive biases that can heavily affect project results. Often seen effects are increased costs and time needed to complete an IT-project. Extant literature has mainly studied the appearance and effects of cognitive biases in traditional waterfall projects. However, focussing on the appearance and effects in waterfall projects leaves the understanding of the bias in other project methods underexposed. As IT-project management methods can heavily differ the assumption exists that biases might also manifest and affect differently. Therefore, this study specifically explores the planning fallacy bias which is defined as: the systematic underestimation of the time needed to complete activities due to an over-optimistic view of reality. This can lead to unrealistic estimates of time, resources, costs and quality and functionality. The study analyses if and how this bias exists in an agile-scrum IT-project setting. An exploratory setup was used, and eighteen agile-scrum experts were interviewed in three different organizations to provide a retrospective insight in the appearance, effects and potential countermeasures. The findings suggest that the planning fallacy bias is present in agile-scrum projects, yet the manifestation is limited compared to traditional IT projects. Iterative planning and optimization ceremonies appear to limit the extent to which the bias affects project results. Yet, the organization maturity in applying agile-scrum seems to be a pitfall. Countermeasures and implications for research are discussed.

Keywords: Software project escalation, decision bias, planning fallacy bias, agile-scrum

Vanwege de complexiteit van IT-projecten wordt de besluitvorming vaak geplaagd door een verscheidenheid aan cognitieve biases die de projectresultaten sterk kunnen beïnvloeden. Veel voorkomende effecten zijn: verhoogde kosten en tijd die nodig zijn om een project te voltooien. De bestaande literatuur heeft voornamelijk de manifestatie en de effecten van cognitieve biases in traditionele watervalprojecten bestudeerd. Door zich te concentreren op de manifestatie en de effecten in watervalprojecten, blijft het begrip van de het effect van biases in andere project methodes onderbelicht. Aangezien IT-projectmanagementmethoden sterk kunnen verschillen, bestaat de aanname dat biases zich ook anders kunnen manifesteren en de resultaten in meer- of mindere mate kunnen beïnvloeden. Daarom onderzoekt deze studie specifiek de planning fallacy bias, die wordt gedefinieerd als: de systematische onderschatting van de tijd die nodig is om activiteiten te voltooien door een overoptimistisch realiteitsbeeld. Dit kan leiden tot onrealistische schattingen van tijd, middelen, kosten en kwaliteit en functionaliteit. De studie analyseert of en hoe deze bias bestaat in agile-scrum IT-projecten. Er werd een exploratieve opzet gebruikt en achttien agile-scrum-experts werden geïnterviewd in drie verschillende organisaties om op basis van de ervaring van de respondenten inzicht te krijgen in de manifestatie, de effecten en mogelijke tegenmaatregelen. De bevindingen suggereren dat de planning fallacy bias aanwezig is in agile-scrum projecten, maar de manifestatie is beperkt in vergelijking met traditionele IT-projecten. Iteratieve plannings- en optimalisatieceremonies lijken de mate waarin de bias de projectresultaten beïnvloedt, te beperken. Toch lijkt de volwassenheid van de organisatie bij het toepassen van agile-scrum een valkuil. Tegenmaatregelen en implicaties voor onderzoek, praktijk worden besproken.

Keywords: Software project escalatie, decision bias, planning fallacy bias, agile-scrum

Summary *[English]*

This thesis assesses the role of the planning fallacy bias in agile-scrum IT projects. Specifically, the appearance and effects of the planning fallacy bias and possible countermeasures are studied. The planning fallacy bias is defined as the systematic underestimation of the time needed to complete activities due to an over-optimistic view of reality. This can lead to unrealistic estimates of time, resources, costs and results. Extant literature studies have shown that cognitive biases regularly cause individuals to fail to deliver functionality on time in full and against the planned budget. A number of studies have provided insight into the role of different biases in waterfall IT projects. Yet the appearance of the biases and its effects in other IT project management methods is understudied. Therefore, this study aims to provide specific insight in the planning fallacy bias in an agile-scrum IT project setting. Agile-scrum has won popularity over the past decade due to the flexibility in project planning. In this context, the agile-scrum methodology has the potential to alleviate the challenges around on-time, in-budget delivery of IT projects. Identifying the appearance and effects of the planning fallacy bias and available countermeasures to prevent appearance will aid in the optimal organizational adoption of the agile-scrum methodology.

Based on an extant literature review understanding was built of the earlier found manifestation and effects of the planning fallacy bias in waterfall IT projects. To enhance the understanding of the bias in the agile-scrum context exploratory interviews were performed. The respondents were all experts in applying the agile-scrum methodology in IT projects. During the interviews a semi-structured questionnaire was used to address the themes manifestation, causes, effects and countermeasures of the planning fallacy bias in agile-scrum IT projects. The interview data was processed and analysed to allow extraction of study results.

The results suggest that the planning fallacy bias is present in agile-scrum projects, yet the manifestation is limited compared to traditional IT projects. Study results show that the possibility to intervene and adjust the planning as the project progresses reduce the effects of the bias. The level of experience of the scrum teams in applying the methodology is found to be a key factor in reducing negative effects of the bias: in experienced agile-scrum teams the bias manifests and affects a project to a very limited extent. An experienced team often applies the agile-scrum methodology more effectively and so facilitates structural refinement of planned activities, reflecting on past development and actively implementing optimizations in the planning accuracy. Based on these findings it is concluded that the agile-scrum methodology can reduce the effects of the planning fallacy bias, yet the organizational agile-scrum experience seems to be a key requirement for reducing the effects.

To validate the results presented in this study future research should focus on measuring the planning fallacy bias in experienced agile-scrum organizations. Additionally, the setup of the research could be improved by focusing on shared experiences among the participants. This setup will allow validation of experiences of one interviewee with those shared by another interviewee and so aid internal validity of the research.

Samenvatting *[Nederlands]*

Deze thesis onderzoekt de rol van de planning fallacy bias in agile-scrum IT-projecten. De manifestatie en effecten van de planning fallacy bias en mogelijke tegenmaatregelen worden bestudeerd. De planning fallacy wordt gedefinieerd: de systematische onderschatting van de tijd die nodig is om activiteiten te voltooien door een overoptimistisch realiteitsbeeld. Dit kan leiden tot onrealistische schattingen van tijd, middelen, kosten en resultaten. Bestaande studies hebben aangetoond dat cognitieve biases er regelmatig voor zorgen dat individuen de functionaliteit niet op tijd, en tegen het geplande budget opleveren. Een aantal onderzoeken heeft inzicht gegeven in de rol van verschillende biases in traditionele waterval IT-projecten. Echter is de rol die biases spelen in andere projectmanagement methoden nauwelijks bestudeerd. Daarom beoogt deze studie inzicht te verschaffen in de planning fallacy bias in een agile-scrum IT-projecten. Agile-scrum heeft het afgelopen decennium aan populariteit gewonnen vanwege de flexibiliteit in projectplanning. In deze context heeft de agile-scrum-methode het potentieel om de uitdagingen rond de tijdige levering van IT-projecten binnen het budget te verlichten. Het identificeren van hoe de bias zich manifesteert, wat de effecten zijn en wat mogelijk aan tegenmaatregelen kan worden gedaan om het effect van de bias te reduceren zullen helpen bij de optimale adoptie van de agile-scrum-methode. Op basis van het literatuuronderzoek werd begrip opgebouwd van de eerder gevonden manifestatie en effecten van de planning fallacy bias in traditionele-IT-projecten. Om het begrip van de bias in de agile-scrum context te vergroten, werden verkennende interviews afgenomen. De geïnterviewde waren allemaal experts in het toepassen van agile-scrum in IT-projecten. Tijdens de interviews werd een semigestructureerde vragenlijst gebruikt om de thema's: manifestatie, oorzaken, effecten en tegenmaatregelen van de planning fallacy bias in agile-scrum IT-projecten te bespreken. De interviewgegevens werden verwerkt en geanalyseerd om tot studieresultaten te komen. De resultaten suggereren dat de planning fallacy bias zich manifesteert in agile-scrum-projecten, maar dat de manifestatie beperkt is in vergelijking met traditionele IT-projecten. Onderzoeksresultaten tonen aan dat de mogelijkheid om in te grijpen en de planning aan te passen naarmate het project vordert, de effecten van de bias vermindert. Het ervaringsniveau van de scrumteams bij het toepassen van de methode blijkt een sleutelfactor te zijn bij het verminderen van het effect van de bias: in ervaren scrumteams lijkt de bias zich nauwelijks te manifesteren. Een ervaren team past de agile-scrum-methode vaak effectiever toe en faciliteert zo structurele verfijning van geplande activiteiten, reflecteert op eerdere ontwikkelingen en implementeert actief optimalisaties in de nauwkeurigheid van de planning. Op basis van deze bevindingen wordt geconcludeerd dat de agile-scrum-methodologie de effecten van de planning fallacy bias kan verminderen, maar dat de volwassenheid van de organisatie een belangrijke vereiste lijkt om de effecten te verminderen. Om de resultaten van deze studie te valideren, zou toekomstig onderzoek zich moeten richten op het meten van de misvatting van de planning in volwassen agile-scrum-organisaties. Daarnaast kan de opzet van het onderzoek worden verbeterd door te focussen op gedeelde ervaringen tussen de deelnemers. Deze opzet maakt validatie mogelijk van ervaringen van de ene geïnterviewde met die gedeeld door een andere geïnterviewde en draagt zo bij aan de interne validiteit van het onderzoek.

Table of Contents

1. Introduction

1.1.Problem definition

Almost half of all IT projects are impacted by poor decision making, individual biases and heuristics (Project Management Institute, 2015). Heuristics are simple decision rules that have led to acceptable outcomes in the past (Kahneman, 2012). Even though heuristics may have been useful in the past, they do not always apply to current or future situations. Unfortunately, individuals find it difficult to identify when heuristics do not apply. When heuristics do not fit the circumstances, this can lead to systematic errors in our thinking and reasoning called biases. Aim of this research is to study the planning fallacy bias (Keil, 2000).

Kahneman describes the bias as the systematic underestimation of the time needed to complete activities due to an over-optimistic view of reality (Kahneman, 1979). Extant research shows that IT projects are prone to the planning fallacy bias because of their complexity due to duration, cross-functional setup, interdependencies and substantial capital investment (Buhl & Meier, 2011). Most studies involve traditional IT projects. As the complexity of IT projects is not reduced by using agile-scrum this method may be prone to the bias as well.

The lack of knowledge on how the bias manifests in agile-scrum IT projects presents a risk, as the bias could unconsciously become a blind spot and negatively impact project results. As the agile-scrum methodology differs from traditional waterfall in multiple aspects, primarily related to planning, it is vital to fill the existing knowledge gap on the impact of the planning fallacy bias (Pinchovski & Fink, 2020). The assumption exists that the bias might appear to a lesser extent as in agile-scrum IT projects as the method requires a short, iterative form of planning. Therefore, the problem statement is:

> “There is insufficient understanding of if and how the planning fallacy bias manifests in agile-scrum IT projects and what differentiates the bias and its effects from traditional waterfall IT projects”.

1.2.Scope and research questions

This study aims to research the role of the planning fallacy bias in agile IT-project management by conducting qualitative, exploratory research (Yin, 2018). The role of the bias in agile-scrum IT projects is defined as the appearance, cause, effects and possible countermeasures.

The following central research question (CRQ) and sub-questions are drafted:

> "What is the role of the planning fallacy bias in agile IT projects?”

1. What is the planning fallacy bias, and how does the bias work?
2. How can the planning fallacy bias be measured?
3. What effect can the planning fallacy bias have on time, costs, functionality and quality of IT projects?
4. What factors can cause the planning fallacy bias?
5. What is the difference between the planning fallacy bias in agile IT projects in comparison to waterfall IT projects?

The five sub-questions will be addressed to answer the CRQ. The first two sub-questions will be answered by performing an exploratory literature study on existing knowledge around how the planning fallacy appears and how it can be measured. This contributes to the CRQ as general understanding is built up on the planning fallacy bias and how it is historically studied and measured. The third and fourth sub-questions contribute to the CRQ by building understanding on the manifestation, cause and effect of the bias, as well as countermeasures that have been developed.

This historic literature study on traditional waterfall projects is combined with semi-structured interviews with respondents experienced in applying the agile-scrum methodology in IT projects. The interviews focus on understanding experience of respondents on the manifestation, cause, effects and countermeasures of the bias. The interviews help to answer the final sub-question. This sub-question is not core to the CRQ but will provide insight into the differences in impact of the bias using different project management methodology. Understanding the difference will increase the practical relevance of this study for organizations and allow transparent insight to consider- or improve application of agile-scrum.

1.3.Project management methods: Waterfall and Agile

1.3.1 Traditional waterfall methodology

The leading traditional project management method within waterfall is PRoject IN Controlled Environment (Prince2). This is a process-driven method that defines seven different project phases from initiation to final closing and decommissioning of the project (Pawar & Mahajan, 2017). The main characteristic of this method is that project planning is made at the beginning for the duration of the project. Once project planning is approved by the steering committee, it is not due to be changed. Defined milestones and deliveries are there to stay (Atkinson, 1999).

Due to the high level of commitment to planning, the method stands out in reliability. Project phases are divided and planning of deliverables is clearly defined and documented. Another characteristic of the methodology is that the decision power lies with the steering committee. This committee usually consists of senior stakeholders in the organization(s) involved. The project manager is the linking pin between the steering committee and the project team and is accountable for the execution of the planned activities.

The high-level planning with rigid conditions and limited flexibility is found to enhance the manifestation and effects of the planning fallacy bias. High-level planning may be prone to be unrealistic, yet the method does not allow to divert from the pre-set milestones and deliveries (Kuhrmann et al., 2017). Therefore, it is seen that projects often do not finish on time and/or additional funding is needed to finish.

1.3.2 Agile-scrum methodology

As an alternative to waterfall, agile project management has been developed (Fowler & Highsmith, 2001). Scrum is a form of agile where in short iterations (sprints) different steps are taken in designing, coding, testing and implementing. It is mainly characterized by flexible delivery of content according to the market or customer needs. The delivery is done by a local, cross-functional and self-organizing team consisting of no more than six members. The development is done through frequent revisions based on lessons learned going forward (Azanha, 2017). As design is not set-in stone, it can be changed between iterations to ensure that customer needs are met.

A scrum-team consists of a scrum-master, who is responsible for facilitating the development process, and the team members, i.e., software engineers who build the functionalities (Schwaber, 2004). The Product Owner (PO) is the end-user and is also part of the scrum team. The PO can decide what the development priorities are and how the development backlog is set up (Shastri, 2017). Based on the complete backlog, the development team decides how much functionality can be developed in one sprint (Azanha, 2017). After a sprint the development team demos the results, the sprint is evaluated in a retrospective, and a next sprint starts.

An advantage of scrum methodology lies in the agility of short iterations used to rapidly develop value for the end-user with room to revise the planning or development direction. This different setup is expected to result in a different role of the planning fallacy bias. As planning is flexible and set for a short period of time it is plausible that the bias has a reduced effect. A disadvantage of scrum lies in the high level of commitment that is needed from the team members as they work together solely on one development phase at a time (Shastri, 2017).

Understanding how both project-management methods work and how they differ is a first step towards understanding how the planning fallacy bias manifests and effects project results in these environments. All learnings will be used to substantiate the first research question.

1.4. Relevance

This study will add to the body of knowledge of decision biases in IT projects by gaining understanding on the role of the planning fallacy bias in agile-scrum IT projects. If the study results reveal that the bias does not appear at all or to a limited extent when using the agile-scrum method, it will aid to understanding the bias. Also, it will pave the way for increased adoption of the agile-scrum method across organizations. If the bias does appear when applying agile-scrum, it is key to understand how it manifests and what can be done to mitigate the manifestation by implementing the found countermeasures.

As the study is performed at three organizations, the results of the research can be compared amongst the different organizations to validate identified patterns. A better understanding of the planning fallacy bias will help the involved organizations to recognize the bias and to take countermeasures to prevent the bias from affecting the results of agile-scrum IT projects.

Finally, identifying how the role of the bias differs in agile-scrum versus traditional waterfall projects will allow a transparent insight in the pros and cons of applying the agile-scrum methodology. Especially for organizations under risk of the effects of the planning fallacy bias it is particularly important to understand how agile-scrum might alleviate this risk. Understanding the difference in how the bias appears is key to substantiate whether agile-scrum IT projects are less prone to the planning fallacy bias.

1.5. Thesis structure

This thesis is structured as follows to examine the stated research question. In chapter two the theoretical background and framework of this thesis are defined. This entails subsections with a description of the literature review and findings. The third chapter contains a description of the research design and methodology. Here, the research approach and methodological choices made for this thesis are displayed. Next, chapter four expounds the results and analysis of the study. In chapter five the most relevant findings are discussed, related to the research and sub-questions, and put in perspective. Further, this chapter includes the practical and theoretical implications, the limitations of this thesis and provides an outlook for further research.

2. Theory

The first section of this chapter will explain the research approach and execution method. In the subsequent section, the IT-project escalation concept and related biases are described. The planning fallacy bias within the scope of this research is set out in section 2.3 to 2.5. Section 2.6 to 2.8 explain how the bias appears and what differentiates the effects of the bias in both traditional and agile-scrum IT projects in terms of time, costs, functionality and quality. The measurement methods will be discussed in section 2.9 and the chapter will be concluded in 2.10.

2.1 Research approach and execution

A literature study lays the foundation for answering the research questions. To reduce influences of construct validity and enable triangulation, both primary and secondary literature sources like professional journals, refereed academic journals and topic-related books will be used. The criteria and parameters used in the literature study are:

A. Keywords: IT-project escalation, Software-project escalation, Decision bias, IT-project escalation AND Decision bias, Planning fallacy AND Agile-scrum, Optimism bias AND Agile-scrum, Planning fallacy AND Waterfall, Planning fallacy bias AND Effect

B. Language of publication: both English and Dutch

C. Geographical area: globally

D. Criteria for citing: the more often an article is cited by others is taken into account in the selection of literature

E. Literature type: refereed journals, books, professional journals, conference proceedings, reports, thesis

F. Databases used: Google scholar is used as the main database as it consists of literature from almost all of the fifty-two available databases. If the literature study with the use of Google scholar does not bring substantial literature the use of other databases will be described and explained

During the research different keywords are used as search terms. The searches led to results, literature reviews and finally, a selection of literature to be used in this study. The search terms were used in the databases, and the sorting was set to 'most relevant':

Table 1 Keywords used for literature review

Search term	**Results**	**Reviewed**	**Selected**
IT-project escalation	154	30	7
Software project escalation	57	12	1
Decision bias	442.000	37	8
Software project escalation AND decision bias	72	8	2
IT project escalation AND decision bias	72	8	2
Planning fallacy bias	102.000	26	7
Optimism bias AND agile scrum	905	7	1
Planning fallacy AND agile scrum	719	23	3
Planning fallacy AND waterfall	1790	32	3
Planning fallacy AND effect	79	7	5

2.2 Escalation and biases in IT projects

Many examples are known of organizations that invest millions of euros in software-development and implementation projects that are endlessly delayed with escalating costs, time and delivery (Keil M., 2010). Drivers of project escalation are often described in:

- Project factors (e.g., long-term investment required with high abandonment cost);
- Psychological factors (e.g., strong personal attachment to project and fear of failing);
- Social factors (e.g., fear to lose face, rewards for staying the course);
- Organizational factors (e.g., side bets, influential sponsors).

This research focuses on the psychological factors because based on historic literature the planning fallacy bias is found in this area of project escalation (Keil M., 2010). A framework that is often used to describe the phycological factors is called the dual-process framework (West & Stanovich, 2001). The two systems described in the framework are:

1. System one that is voiced over as our intuition;
2. System two that is voiced over as how we reason.

IT projects are prone to biases because of the complexity due to the duration, cross-functional setup, interdependencies and substantial capital investment and budgets to be managed (Buhl & Meier, 2011). The combination of the complexity and uncertainty of development speed and potential challenges overloads system two for processing and reasoning and so individuals fall back on their heuristics.

The planning fallacy bias refers to the systematic underestimation of time needed to complete future activities. It appears individuals find it hard to consider 'what if' scenario's when planning activities. The bias can lead to inaccurate project planning, inadequate risk management and shortfall in delivery (Flyvbjerg, 2006). Even though the bias can affect project results tremendously, there is very little research and understanding of how the bias manifests in agile-scrum IT projects.

2.3 The planning fallacy bias in traditional IT projects

Manifestation

The planning fallacy bias is often triggered in situations where abstract, high-level construal planning is performed (Shalev, 2014). When creating a project and risk plan, individuals tend to be over-optimistic in regard to the end goal and project benefits and often underestimate costs and efforts to achieve it. The planning fallacy is relevant in any context to time overrun, cost overrun, over-scoping functionality and underdelivering quality (Shmueli, 2016).

Causes

One of the causes found for the bias is the human tendency to disregard previous experiences in corresponding activities, taking an 'inside view' rather than an ‘outside view’. The views differ in whether a problem or forecast is seen as unique or as part of a cluster of similar activities planned in the past (Kahneman et al., 2003). The inside view can cause the bias as experienced learnings are not clustered and so negative experiences can easily be disregarded (Buehler et al., 1994). This view affects IT projects that traditionally rely heavily on detailed planning and forecasting (Kahneman & Lovallo, 1993).

Effects on time and costs

The planning fallacy bias is often found to affect costs and time in traditional IT projects. Cost-overrun is the difference between forecasted and actual cost (Sing, 2013). Time-overrun is caused by team members underestimating the time needed to complete activities. Participants in IT projects tend to exaggerate the role of control and their capability to predict future results (Kahneman et al., 2003). This often leads to an underestimation of the time needed to complete the planned activities just as an underestimation of the budget and resources needed to execute them (Charette, 2010). Especially in traditional IT projects this underestimation of time needed can enhance the manifestation of the planning fallacy bias. As planning is not subject to change, one mistake in the planning phase can have long-lasting effect and additional budget or resources might be needed to be able to complete the activities on time.

Effects on quality and functionality

Scope-overload describes the inclusion of more software features within the scope of a project than practically possible in the available time, resources and budget (Bjarnason, 2012). Scope-overload is studied in relation to the planning fallacy bias and has shown impact and consequences on quality and functionality, leading to failing customer expectations (Buschmann, 2010). Project team members tend to include too much functionality than the project scope allows (Buschmann, 2010). In traditional IT projects scope is often defined in an early project initiation phase, but usually, after the project duration is defined. Early scope definition is prone to the planning fallacy bias as it proves to be challenging to estimate the time needed to complete activities accurately. It can quickly result in over-scoping and making on-time delivery unrealistic from the start (Nouri et al., 2012).

Relevance

Understanding the manifestation, cause and effects of the planning fallacy bias on cost and time overrun and of scope and functionality overload helps to substantiate the effects of the bias corresponding with research question three. Understanding the effect of the bias in the historic studies of traditional IT projects helps to understand the role of the bias in general.

2.4 The role of the project management method

Traditional and agile-scrum IT projects differ in how planning is performed. Traditional IT projects tend to plan in a rigid manner: before the start of a project a big number of interdependent features and deliverables are planned for period of months to years (Sanna & Chang, 2005). Agile-scrum IT projects tend to plan short in iterations of a limited scope of features and deliverables (Usman & Mendes, 2014). As the planning in agile projects is done for short periods, the effect of the planning fallacy bias could be significantly less. The interdependencies between different activities might be more apparent (Sauer, 2007). The comparison of both methods allows validation of the assumption that agile-scrum is less prone to the bias.

2.5 The planning fallacy bias in agile IT projects

Manifestation

Even though there is limited high-level planning done in agile-scrum IT projects the bias might still manifest (Brown, Dirska, & Pelosi, 2013). Yet, as planning is done in shorter iterations and evaluation takes place after every evaluation the appearance of the bias might be identified early on and tackled easier. Also, the impact of the bias might be reduced as planning only affects one or two iterations and not the complete project. Whilst understanding this it is likely that still individuals have difficulty to realistically plan complex work even when planning in short iterations (Shmueli, 2016).

Causes

The found causes such as the human tendency to disregard previous negative experience and to plan activities overly optimistic still exist in agile-scrum projects. Even if the planning scope is much smaller and the risk of self-blame due to failing individuals still come to find it challenging to accurately plan complex activities (Shmueli, 2016). Yet the effect of the causes on the final project results might be significantly reduced (Brown, Dirska, & Pelosi, 2013).

Effects on time and costs

Overrun of costs and time is often a result of failed planning of activities (Shmueli, 2016). In agile-scrum, budget and time are taken as fixed variables while the functionality is a flexible variable (Bogdan, Cosmin, Sorin, & Costin, 2019). Therefore, it might be the case that less functionality is delivered, but the planned deadline and budget are appreciated. Furthermore, agile-scrum might be affected differently than traditional Prince2 because of the difference in planning flexibility. Planning is made only a few sprints ahead and might, to a lesser extent, impact the cost and time-overrun if they appear at all (Usman & Mendes, 2014).

Effects on quality and functionality

The impact of the bias on quality and functionality might be more prominent. As time and cost are fixed variables, the implications of inaccurate planning will be on the delivered quality and functionality (Bogdan, Cosmin, Sorin, & Costin, 2019). While assuming that the bias is still at play, too much functionality will be planned in a sprint which makes delivery not realistic in time and budget. This unrealistic scope will lead to iterative development cycles scaling down the functionality and quality to meet time and cost restrictions.

Relevance

Understanding the historically identified manifestation, cause and effects of the bias in agile-scrum IT projects helps to answer the first and fourth research question and helps to form a substantiated assumption on the role of the bias in agile-scrum IT projects.

2.6 Measuring the planning fallacy bias

In previous research multiple methods are identified to measure the bias. Studying how other researchers have measured the planning fallacy bias is interesting as it might provide a basis for measuring the bias in this research.

One method of measuring the bias is by running real-life experiments with groups of individuals testing their forecasting accuracy by handing over cases and mapping or questioning the individuals about what choices they would make to forecast and plan. After completion of the forecast the individuals will actually execute the planned activities and inaccuracies can be measured by comparing the forecast with the actual situation (Buehler, Griffin, & Ross, 1994). Another method is measuring the bias by interviewing individuals about their experience on agile-scrum IT projects to determine the role of the bias on the project. In the interviews open-ended questions are used and the results of the interviews are transcribed and coded to find patterns in the results that allow drawing conclusions.

Focusing on this research the interview method with use of open-ended questions is selected. Due to limited availability of the respondents the interviews are a better fit as they can be planned at the best suitable time for the respondent. Respondents are asked to share their experience on how they planned versus how a project actually went. This comparison, drawn by different respondents, presents insight in whether and how the bias manifested in these real-life IT-project environments and how it affected project results. The interview method is therefore preferred over the experiment method.

2.7 Results and conclusions

The planning fallacy refers to the systematic underestimation of the time needed to complete activities due to an over-optimistic view of reality. This may lead to an overrun of projected costs and time or to under delivery of functionality and quality that may cause a project to fail. Historic research focuses on the appearance of the bias in traditional IT projects and hardly on the increasingly popular agile-scrum IT projects. The known effects of the bias might differ in agile-scrum IT projects as individuals can plan in shorter iterations with more flexibility to the planning. This discrepancy can reduce the identified causes of the bias regarding fear of failure and therefore overselling planning and or underestimating risks.

2.8 Continued research direction

Understanding the bias in traditional IT projects is critical before setting the next steps in understanding the bias in agile-scrum IT projects. Chapter three will continue developing the research method to build the foundation for interviews with respondents experienced in agile-scrum. Chapter four will provide an analysis of the interview data and finally the conclusion and recommendations for future research will be shared.

3. Research method

This chapter explains the research method and design. In the first section, the selected conceptual model is described. The subsequent two sections elaborate on how the method is operationally used and how data analysis is conducted. In the final subsection, a reflection on the process is shared.

3.1 Conceptual design: Method selection

To obtain a broad understanding of the bias and its effects scientific literature was used. As there is limited research on the planning fallacy bias in agile-scrum projects, primary data was additionally collected through respondents who actively work in agile-scrum IT projects and have built up experience in this environment.

The research focuses on organizations where a group of individuals are experienced in using the agile-scrum IT-project method. With unique access to a group of respondents with such experience it can be revelatory to use interviews to develop understanding of how respondents planned versus how a project actually went. This comparison, drawn by different respondents, presents insight in whether and how the bias manifested in these real-life agile-scrum IT projects and how it affected project results. In doing so this research aims to explore and develop a theory rather than testing hypothesis.

As the sample size within the organizations is limited and the study seeks to explore and build theory, a qualitative research method is preferred over a quantitative one. Scripts will be used to structure the interviews, and questions focus on themes: manifestation, causes, effects and countermeasures. These themes also form the basis for the coding of the interviews afterwards. Historic literature was used to develop the interview questions (Benschop, Rijsenbilt, & Wilmink, 2020). The questions are presented in Appendix A. The interviews were used to both confirm already found elements from historic studies as well as reveal new elements.

In summary, based on the above rationale: the research will be conducted using a qualitative, exploratory setup where the fundament is formed by a literature study and additionally built upon by interviews.

3.2 Technical design

In the interviews, emphasis lies on the experiences of the respondents regarding the planning fallacy bias in agile-scrum IT projects. For this study eighteen interviews have been conducted in three different organizations by three different interviewers. The respondents were selected on the following criteria:

- Respondents have worked in agile-scrum IT projects and have completed at least two agile-scrum IT projects from beginning to end;
- Respondents have been responsible as either scrum-master, product owner or engineer in an agile-scrum development team, with the aim to select the roles evenly.

Interview materials were prepared to facilitate the collection of the intended qualitative data. In an introduction letter to respondents, the three different biases were presented. This letter provided the name and a brief description of each bias and an illustration associated with the bias, see appendix B. At the start of every interview the descriptions and illustrations were read, and the respondents were asked to indicate whether the bias was clear to them. Two pilot interviews took place to assess if the introduction letter and descriptions and illustrations were understandable and whether the interview protocol was clear. Based on the pilot a few changes were made to the wording of the illustrations and order of the interview questions to make them clearer.

Due to the pandemic circumstances the interviews have been conducted through web conferencing. The interviews lasted between 60 to 90 minutes, in which three biases were discussed following the interview protocol. The protocol allowed standardization of how the interview was conducted across interviewers. All interviews were recorded, and transcripts were made. Privacy legislation has been respected as names and produced data has been anonymized. The data have been stored in an online database that contained the introduction letter, interview protocol, audio files, a spreadsheet with all themes and corresponding codes including scores, quotes and references to specific respondents. Specifically, the spreadsheet allowed analyzing and comparing data across interviews and helped to develop and group codes. The next subsection will further elaborate on the coding process.

3.3 Data analysis

Text analysis was applied to analyze the interview data. Data was split into fragments by coding and organizing into pre-determined themes. The data was reduced by summarizing the meaning to comprehend and perform further analysis while retaining the integrity of the data.

When coding the transcript data, each fragment received a code that summarized the meaning and was focused on four themes: manifestation, cause, effects on time, costs functionality and quality and countermeasures.

The use of codes enables comparison of data fragments from different transcripts and further analysis to identify patterns. The defined themes and codes are leading during the coding process but can be expanded if particular new labels become important during data analysis (Saunders et al., 2015).

3.4 Reflection on saturation, validity, reliability and ethics

To validate whether the input of the eighteen respondents in this study would be sufficient to draw conclusions from the qualitative data, the data saturation was measured. The development of new- and existing codes were measured after every interview to identify at what point the data saturation was reached.

A reliability risk lies in inconsistencies in the interview method as different interviewers are involved. This risk is mitigated by standardizing interview questions and sharing recordings and transcripts to allow evaluation of the reliability afterwards.

A third risk lies in the limited sample size used in the research, which pressures the generalizability of the results. This risk is managed as the research aim is not to present findings that are statistical generalizations, but rather to reflect on actual results in a specific situation at one particular point in time.

A final remark concerns the internal validity as the research concerns historical situations that cannot be checked or retrospectively controlled. This internal validity risk is an effect of this research setup and should be considered during the research and noted in the research limitations.

Ethical aspects have been taken into account in this research. Respondents' consent on participation to the study is requested. Also, anonymity is prioritized from the start, as names and respondent information will be captured in IDs to ensure nothing refers to the respondents' actual identity.

These reflections taken together lead to the conclusion that the data from the 18 interviews provide a sufficient basis for the conclusions drawn in the context of this exploratory research. This exploratory study does not attempt to generalize to a population like many survey studies in this domain do. It is aiming to obtain a more detailed and coherent insight on the effects of the planning fallacy bias on agile-scrum IT projects. The results of the analyses are discussed extensively in the next chapter.

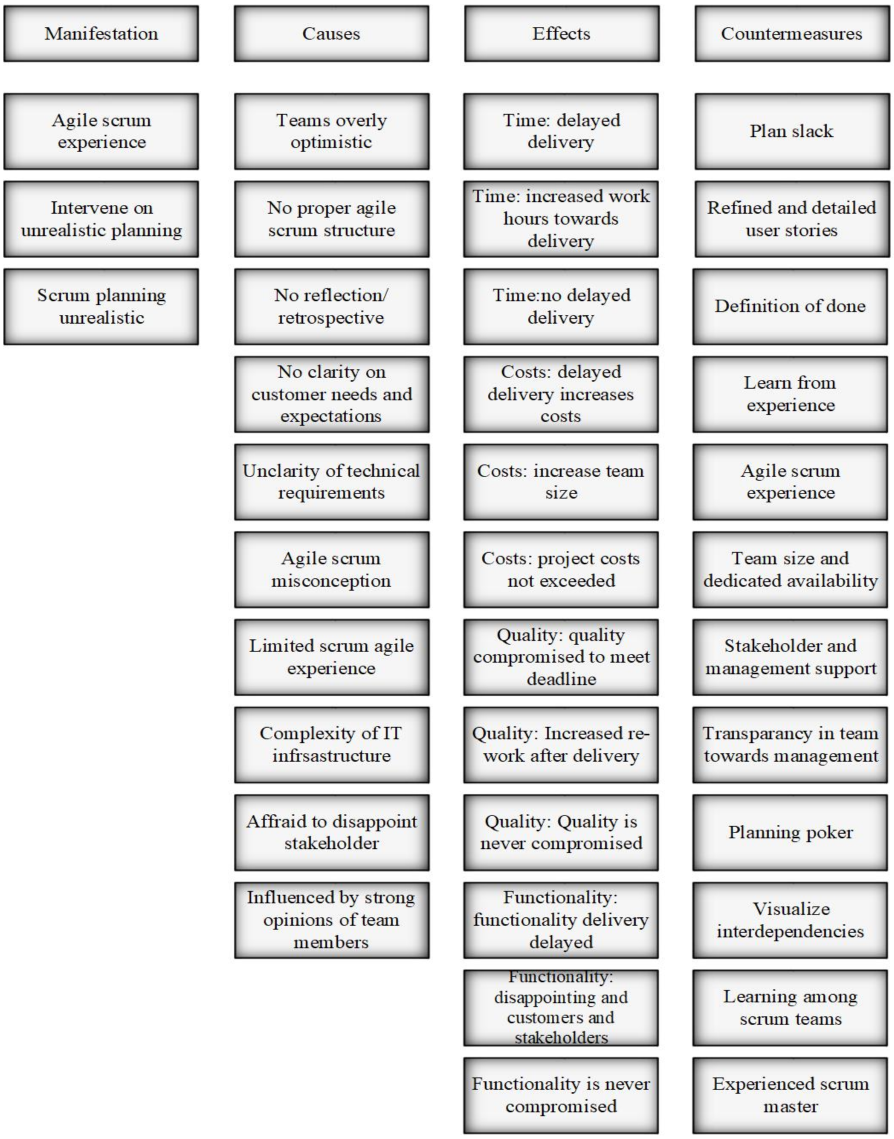
4. Results & Analysis

This chapter presents the findings, based on the research model that has been built in the previous chapter. The central research question aims to develop understanding on if and how the planning fallacy bias manifests in scrum agile IT projects. This chapter will cover the answers to the third and fourth research question:

3. Which effect can the planning fallacy have on planning, budget, functionality and quality of IT projects?
4. Which factors cause the planning fallacy bias?

The results of the semi-structured qualitative interviews and subsequent analysis on the identified themes and related codes are discussed in the sections below. The final section summarizes the key findings and gives an outlook for the discussion.

The complete code schema can be found in Appendix C; it consists of the description of the codes as well as the frequency the codes occur. A summarized version is shown in figure one.



Manifestation	Causes	Effects	Countermeasures
Agile scrum experience	Teams overly optimistic	Time: delayed delivery	Plan slack
Intervene on unrealistic planning	No proper agile scrum structure	Time: increased work hours towards delivery	Refined and detailed user stories
Scrum planning unrealistic	No reflection/ retrospective	Time:no delayed delivery	Definition of done
	No clarity on customer needs and expectations	Costs: delayed delivery increases costs	Learn from experience
	Unclarity of technical requirements	Costs: increase team size	Agile scrum experience
	Agile scrum misconception	Costs: project costs not exceeded	Team size and dedicated availability
	Limited scrum agile experience	Quality: quality compromised to meet deadline	Stakeholder and management support
	Complexity of IT infrsastructure	Quality: Increased re-work after delivery	Transparancy in team towards management
	Affraid to disappoint stakeholder	Quality: Quality is never compromised	Planning poker
	Influenced by strong opinions of team members	Functionality: functionality delivery delayed	Visualize interdependencies
		Functionality: disappointing and customers and stakeholders	Learning among scrum teams
		Functionality is never compromised	Experienced scrum master

Figure 1 Code overview per theme

4.1 Expert-interviews

In the sections below the findings of eighteen interviews are discussed. The interviews were held by three different interviewers, at three different companies operating in different sectors, both private and public. The interview recordings were shared among the interviewers in order to create the transcripts and analysis for the planning fallacy bias interviews. The group of respondent's varies in gender, age (30-65 years old) and role (i.e., scrum master, engineer, product owner), and years of experience. As some of the respondents were Dutch speakers their quotes were translated to English to allow other researchers to understand and potentially build on this research. The quotes are shown when enhancing the understanding of a code.

4.1.1 Manifestation

The "Manifestation" theme describes the overall recognition of the bias in agile-scrum environments, the practical appearance and factors that might increase or decrease the manifestation. The respondents' answers regarding if and how the bias manifests led to three different codes that are discussed below.

Scrum planning unrealistic

This code refers to overestimating the number of activities that can be completed in a specific amount of time. The manifestation code of the bias is found eight times in the interview data and respondents. Respondents emphasized the psychological tendency to overestimate what one can deliver in a certain amount of time. As one respondents describes: *"People really underestimate the time required to complete activities [...] you cannot really estimate complex work accurately".*

Agile-scrum experience

The code refers to the maturity level of the scrum-team in correctly applying the method and the experience in developing in short iterations. Thirteen respondents share that the way and the extent to which the planning fallacy manifests is dependent on the maturity of the agile-scrum-team. The more experienced a team is in applying the methodology, the better and the more accurate a team can predict the time needed to complete activities. A respondent describes: *"It has to do with the experience of the team [...] that they at some point know we are 80% certain we can deliver what is planned"*. In inexperienced teams the planning fallacy bias manifests heavily leading to inaccurate planning and delayed delivery of functionality. Lacking experience will, due to the significant effect, also be described in the "Causes" theme.

Intervene on unrealistic planning

This code refers to the way the agile-scrum methodology limits the manifestation of the planning fallacy bias in the ability to be aware of the bias and adjust the planning iteratively. Ten respondents specifically share that they recognize the manifestation of the bias and connect the extent to which the bias manifests to the intervention opportunities the methodology allows. As the methodology uses short iterations followed by small delivery and evaluation moments, it enables the scrum-team to adjust planning of activities after every iteration.

Theoretic overlap

When comparing the found and analyzed codes to the findings of the literature review it stands out that team maturity plays a very significant role in the extent to which the bias manifests. The ability to timely intervene was already identified and even expected based on previous studies done.

4.1.2 Causes

The "Causes" theme describes the overall identified causes of the bias in agile-scrum environments. Respondents' answers regarding the causes of the bias in agile-scrum project environments lead to eight different codes which are discussed below.

Overly optimistic

This code refers to the human tendency to project an overly optimistic view when estimating the time needed to complete project activities. Fourteen respondents describe that when making a planning often interdependencies are not forecasted directly which later on often heavily delays the project. This accounts for planning over longer periods of time but just as well for planning for shorter iterations.

No agile-scrum structure

This code refers to the incorrect application of the agile-scrum method. A respondent shares that this causes a lack of structure and moments to challenge the planning in the refinement or evaluate during the retrospective which often causes delays in delivery.

No reflection / retrospective

This code refers to the lack of reflection on team performance. Five respondents shared that without proper reflection it is difficult to identify improvement areas, such as more realistic planning of activities, let alone implement improvements to the way of working in the team. A respondent describes: *"If you don't look back, try to understand what went wrong it's really hard to be aware of the bias and act on it".* The respondents noted that in order to reduce the impact of the bias over time it is key to ensure the team is aware of the bias and recognizes inaccurate planning. Based on this awareness an improvement action can be implemented.

No clarity on end-customer expectations

This code refers to limited understanding of the expectations of the end-customer and also a lack of granularity in describing the expectations in the form of a user story. A respondent describes: *"If the specifications are not written down in detail or elaborated on by the end customer then it becomes even harder to predict the effort in time".* Three respondents support this.

In-transparency of technical impediments

This code refers to the inability of a team to predict technical impediments that might arise during the development as what they are doing has never been done before. Nine respondents share that activities are often completely new and so predicting how long it takes to complete is very difficult. Respondents suggest that these in-transparencies can lead to assumption-based planning which is prone to be unrealistic. In agile-scrum this can play a role as the responsibility of providing this transparency lies in the agile-scrum project team and in many cases this responsibility is not fulfilled.

Agile misconception

This code refers to an inaccurate or incomplete understanding of the methodology which can lead to inaccurate expectations. Five respondents describe this. This misconception leads individuals to expect that everything is planned on a granular level and will be delivered on time. The methodology is thus interpreted to be a magic want. When agile-scrum IT-project management is not commonly used and understood within organizations it can cause the bias.

Limited agile-scrum experience

This code refers to the insufficient understanding of applying agile-scrum methodology as a causing factor of the planning fallacy bias. Nine respondents share that due to the lack of experience scrum teams have more difficulty to slice the work in smaller chunks and have the tendency to plan unrealistically.

Complexity of IT-infrastructure

The large organizations that participated in the study have complex IT-infrastructures and the IT projects they execute have an innovative character. Estimating how long it will take to complete planned activities is as much based on gut feeling as on experience. Twelve respondents share that this cause is applicable to scrum agile projects as this project structure is relatively flexible and you can easily divert planning and adjust goals. A respondent shares: *"We are constantly working with new platforms, products and programming languages that have to be connected to legacy systems. Understanding how that needs to be done is new and highly complex."*

Afraid to disappoint stakeholders

Fear of team members to disappoint or disagree with management or senior stakeholders leads to agreeing on deliverables and project planning that is actually not feasible. Agile-scrum IT projects can be influenced by this element as the responsibility and accountability lies in the team. This different allocation of responsibility and accountability in the project team presents a high contrast compared to traditional IT projects. As the traditional IT-project management method was historically used in the studied organizations respondents share that managers easily fall back in old behavioral patterns and try to influence the decision making than the agile-scrum methodology actually prescribes Six respondents describe this and one shares: *"The MT plays a role if they pressure and say 'come on it is not difficult right? You can deliver faster', so you scqueeze until the moment the plan is completely trash."*

Influenced by strong opinions in team

This code refers to situations where an opiniated team member is not opposed in an effort to avoid conflict. The team may agree on a planning that is driven by only one team member while it is not supported by the whole team. A respondent shares that the misalignment in a team driven by strong opinions often leads to unrealistic planning.

Theoretic overlap

The human tendency to plan over-optimistically was already found in the literature review. The same accounts for the complexity of IT-infrastructure and the in-transparency of technical impediments. The seven codes listed above appear to be new elements that were not found in the literature review and therefore can be an addition to the scientific knowledge on which specific elements cause the planning fallacy bias.

4.1.3 Effects

The "Effects" theme describes the overall identified effects of the bias in agile-scrum environments. The respondents' answers regarding the effects of the bias in agile-scrum project environments led to four sub-themes and thirteen codes. The sub-themes and related codes are described below.

Time-effects: delayed delivery

This code refers to a delayed completion of deliverables beyond the initial agreed date. Thirteen respondents share that due to unrealistic planning too much work is planned in a too short amount of time. This makes it hard to deliver everything as initially planned. A respondent shares: *"[...] because you cannot oversee what needs to happen you plan inacurate and for sure deliver things later than planned"*.

Time-effect: increased work hours towards delivery

Five respondents share that when a scrum-team is under pressure to deliver a sprint before deadline although the time available is actually not sufficient, the only way to make ends meet is to spend additional hours above the already forecasted hours.

Time-effect: no delayed delivery

This code refers to the situation where delays do not exist as time is a fixed variable in the agile-scrum project setup. Eleven respondents shared that in applying the methodology time was never compromised over other elements such as functionality or quality. The data shows that in these situations the team can decide to win time by reducing the quality or functionality. Respondents share that it is in the nature of the innovative character of the projects to not know how much functionality can be delivered and therefore this element can be adjusted.

Cost-effect: delayed delivery increases costs

This code refers to the longer utilization of the scrum-team due to the delayed delivery unrealistic planning. Fifteen respondents share that when the planning is unrealistic, and the actual developments takes longer than expected it automatically affects project costs as the scrum-team has to develop longer than anticipated.

Cost-effect: increase team size

Three respondents share that unrealistic planning often requires adding people to the team when the team is heading towards a delayed delivery of functionality. Adding people to the team is identified as a last resort and occurs when there is significant pressure on the on-time delivery of the functionality. This situation occurred especially in the public domain where changes in jurisdiction are often political deadlines tied to release dates.

Cost-effect: project costs not exceeded

This code refers to the situation where delays do not exist as costs are a fixed variable in the agile-scrum project setup. Ten respondents share that the delivery against the planned costs is so important that functionality and quality are sacrificed over costs. A respondent shares: *"costs and time are fixed, output is variable"*.

Quality-effects: quality compromised to meet deadline

This code refers to the tendency in a project to sacrifice quality delivered rather than time, costs and functionality. Nine respondents share that reducing quality of the product is often not as visible for the end user or stakeholder. Therefore, when due to unrealistic planning delivery comes under pressure, quality reduction is a quick win. A respondent shares: "*The front end looks nice but the back end is a mess, not scalable"*.

Quality-effect: increased rework after delivery

This code refers to the situation where due to time pressure on delivery a team sacrifices quality and has to pay the price for reworking after the functionalities are delivered. Three respondents share that often when sacrificing quality, it is difficult to oversee the final effects of this decision. Delivery of reduced quality leads to unforeseen challenges in the further development of the project. Respondents share that the rework, repair and increase of quality later on in the project often takes more time and thus increases development costs.

Quality-effect: quality is never compromised

This code refers to the setup of the agile-scrum method in an organization with such strict quality standards that compromising quality is impossible. Ten respondents share that quality is seen as a fixed variable and costs and time are sacrificed over quality.

Functionality-effect: functional delivery delayed

This code refers to the situation where the delivery of functionality is delayed due to unrealistic planning of activities. Respondents share that when a team is under time pressure due to unrealistic planning caused by the bias, an often-sacrificed element is the functionality. Thirteen respondents describe that within the complex infrastructures of a project the functionality to be delivered is not fixed and can be changed.

Theoretic overlap

Based on the literature review the expectation existed that both time and costs would be fixed variables and functionality and quality would be flexible variables. It stands out that this expectation is not confirmed by the research result as all elements can be fixed or flexible, depending on the organization or project decisions. Therefore, the effects of the bias on the time, costs, quality and functionality are all new. For all the thirteen elements it accounts that the level of detail and rationale found for every element is to an extent revelatory and can add to the scientific knowledge.

4.1.4 Countermeasures

The "Countermeasures" theme describes the measures to address the bias in agile-scrum environments that have been identified. Respondents' answers regarding the countermeasures led to the following twelve codes.

Plan slack

This code refers to structurally planning additional time to complete an activity as a margin of error. Three respondents share that it is very common to underestimate the time needed to complete an activity. Planning additional time allows the team to finish on time without or with less time pressure or spill over from one sprint to the other. When the team does not need the additional time, it can be used to start working on other activities on the backlog.

Refined and detailed user stories & planning poker

This subsection describes two codes connected refinement aspects. The "refined and detailed user stories" code refers to the level of granularity that is shown in the user stories and the level of involvement of the end-customer that is shown in the user story. Thirteen respondents point out that a high level of detail on the functionality to be delivered is key to accurate planning, along with a clear understanding of the customer problem and expectations. If the user story is not captured in detail it is very difficult for the scrum-team to define an accurate planning of all the sub-activities which will allow the bias to manifest increasingly.

"Planning poker" refers to the importance of appointing complexity points to user stories in order to identify how complex the user stories are and so how much time is needed to complete the user stories. Four respondents share that a poker session allows the team to discuss all user stories, identify if the stories are complete and if so, assign complexity points to the user story to allow an accurate planning.

Definition-of-done

This code refers to the availability of a clear and measurable description of what needs to be delivered. Having a definition-of-done available will, according to four respondents, allow the scrum-team to quickly understand when a user story meets the criteria that are set for marking it as done. These criteria will ensure that the team will not overdeliver on quality or functionality aspects and allows an effective use of the time available in a sprint.

Learning from experience & learning among scrum teams

This subsection describes two connected codes in the learning aspect. The first code, "learning from experience", refers to the structural evaluation of a sprint and active implementation of optimizations. Eleven respondents share that leveraging a retrospective can enable the team to make a more accurate estimate of the time needed to complete activities and to identify actions to increase performance.

The second code, "learning among scrum teams" refers to the structural exchange of learnings on planning accuracy to avoid the same mistakes to be made by different teams. Two respondents share that learnings from one team can be implemented in other teams. The bias will be mitigated when learnings of mature teams provide a powerful insight for less mature teams.

Agile-scrum experience & experienced scrum-master

This subsection describes two codes connected in the experience aspect. The "agile-scrum experience" refers to the maturity level of the team based on experience in applying agile-scrum methodology in IT projects. Four respondents share that experienced teams will deliver a more accurate delivery planning. The experience helps in understanding and flagging potential impediments in the refinement process and anticipating on these impediments when planning

The "experienced scrum-master" code refers to the maturity of the scrum-master in correctly applying the agile-scrum methodology. Three respondents share that a mature scrum-master facilitates the team to perform optimally by challenging time planned for completing activities and identifying and measuring solutions.

Team size and dedicated availability

This code refers to the importance of having a sufficient team size and have the dedicated availability of the team members. A respondent shares that when underestimating planning often also the resource need is underestimated. Additionally, team members often work across different projects at the same time which can cause delay in availability. Having the right team size and dedicated availability ensures that even when planning is underestimated the team members are available to deliver on time.

Stakeholder and management support & transparency towards management and stakeholders

This subsection describes two codes connected in the stakeholder management aspect. “Stakeholder and management support” refers to the importance of supportive stakeholders and managers who serve rather than pressure the team and allow it to make mistakes in order to learn. Six respondents share that pressure from managers or stakeholders often pushes scrum teams to commit to unrealistic planning and enhances the manifestation of the bias.

The “transparency towards stakeholders” code refers to the importance of creating transparency on what is in development, what impediments are identified and what the next steps are. Four respondents describe that providing this transparency allows stakeholders and managers to build trust and confidence in the performance of the team. It will limit the tendency to push the team or become overly involved in team decisions.

Visualize interdependencies

This code refers to the importance to visualize the planned activities and visualize where and when interdependencies exist between planned activities. Two respondents share that visualizing interdependencies between planned activities helps to anticipate and to open up a constructive discussion on managing them. Being aware of the interdependencies and discussing these in scrum teams or among team members allows more accurate planning.

Theoretic overlap

The codes regarding refined user stories, sprint evaluation, team size and dedicated availability have been identified in previous studies. Yet the level of detail and extensive description of these specific elements and how these countermeasures can mitigate the bias enhance the existing knowledge. The elements refinement and evaluation are found to be part of the agile-scrum process, but the codes show that the application of the process often differs in practice. This potentially invites the risk of manifestation of the bias. New elements found are the codes regarding planning of slack, definition of done, agile-scrum experience and scrum-master experience, stakeholder management and transparency to management, visualize interdependencies. Understanding these elements will allow to build a better understanding of the planning fallacy bias in agile-scrum IT projects.

4.2 Theoretical saturation

A graph visualizing the interview saturation is shown in Appendix D. To validate whether the input of the eighteen respondents in this study would be sufficient to draw conclusions from the qualitative data, the data saturation was measured. When processing the interview data, the new number of codes were tracked after each interview. The data showed that all codes were gathered in the first ten interviews, the following eight interviews only confirmed the already identified codes but did not identify new codes. Based on this insight it is concluded that this analysis shows that saturation is reached after the eighteen interviews taken in this study.

4.3 Summary of the findings

This section addresses both the third and the fourth research question. The interview results were analyzed, and the noteworthy points were divided into themes and codes, based on the literature review. The primary findings show that the planning fallacy bias does manifest in agile-scrum IT projects. Even when planning activities for shorter periods of time still planning is described to often be inaccurate. Yet, findings show that the impact of the planning fallacy bias in agile-scrum IT projects is limited. Teams seem to become aware of the inaccurate planning due to the short iterative character of the agile-scrum method and can easier adjust planning and so improve the reliability of their delivery. A remark is made as the experience level of agile-scrum is found to be crucial in reducing the effect of the planning fallacy bias. The less experience exists in a team, the heavier the effect of the bias on project results. Finally, there are countermeasures identified that can be implemented to reduce or even mitigate the effect of the bias whilst allowing teams to develop their experience in agile-scrum.

By scrutinizing the two sub-questions the manifestation, causes, effects and countermeasures of the planning fallacy bias in agile-scrum IT projects are broadly established. Understanding of the bias in agile-scrum IT projects opens an interesting pathway forward for building further understanding of the bias in agile-scrum. This outlook will be discussed in the next chapter.

5. Discussion, conclusion and recommendations

The following sections elaborate and conclude on the key findings. These sections are followed by a reflection on the research and the implications for practice and researchers. The final section describes the recommendations for future research.

5.1 Discussion of key findings

This thesis aims to study the role of the planning fallacy bias in agile IT projects by conducting qualitative, exploratory research (Yin, 2018). The role of the planning fallacy bias refers to how the bias manifests, what causes the bias, how it effects time, costs, quality and functionality and what countermeasures can be taken to reduce the effects of the bias in agile-scrum IT projects. The literature review showed that the planning fallacy bias has been studied for decades, often focusing on the traditional waterfall IT projects. In these settings it is acknowledged that the planning fallacy bias is triggered by the abstract, high-level construal planning and often leads to an overrun of time and costs as well as an under delivery of functionality and quality (Shalev, 2014) (Shmueli, 2016). This exploratory study contributes by enhancing the understanding of the planning fallacy bias in the specific agile-scrum IT-project setting. The main findings are discussed per theme hereafter.

Manifestation of the planning fallacy bias in agile-scrum IT projects

Interviewees share that the manifestation of the bias is generally limited due to the short iterative character of agile-scrum. Yet, they share that the experience level of a scrum-team in applying the methodology can heavily affect the extent to which the bias manifests. A higher level of experience is found to correspond with limited effect and so limited manifestation of the bias, and vice versa.

Causes of the planning fallacy bias in agile-scrum IT projects

Limited experience in applying agile-scrum and limited understanding of what it entails are found causing factors of the bias. The limited understanding of the method leads to a sub-optimal execution of the refinement and retrospective sessions which are crucial in agile-scrum. Additionally, the highly complex IT-environments and the innovative character of IT projects often allow the bias to manifest. are Finally, the excessive influence of stakeholders or dominant team members can lead to agreeing to an unrealistic planning.

Effects of the planning fallacy bias in agile-scrum IT projects

The study results reveal that the effect of the bias on time, costs, functionality and quality differ depending on the organizational decision in this respect. Where organizations decide to set time and costs as fixed variables it stands out that the planning fallacy bias triggers over-scoping of functionality and quality and makes delivery against time and budget unrealistic (Bogdan, Cosmin, Sorin, & Costin, 2019). Additionally, the reduced delivery of especially quality is stipulated to lead to higher costs later on as additional re-working has to be completed, taking away time and resources to work on new developments. When functionality and quality are set as fixed variables, the planning fallacy bias leads to delayed delivery against higher costs. The effects of the bias often cause disappointment among stakeholders, management and end users.

Countermeasures of the planning fallacy bias in agile-scrum IT projects

The following countermeasures have been identified to reduce or mitigate the manifestation of the bias:

- Correct application of agile-scrum is broadly identified by respondents. Doing so will primarily ensure that user stories are clearly defined and understood by the development team which allows accurate planning and increased delivery reliability.
- Allocating complexity points to the activities and discussing this allocation with the scrum-team helps to estimate planning for one activity in comparison to another and further increases planning accuracy.
- Performing the prescribed evaluation of completed activities will raise awareness if unrealistic planning is systematically done and allow implementation of improvement actions.

- Ensuring that a sufficient experience level of a team is found to be a critical measure to be aware of.
- Transparency on the planned activities and impediments and sharing these insights with stakeholders makes it less likely for stakeholders intervene in the team as they know the status and feel confidence in the capabilities of the team to deliver.
- Finally, planning additional time as a margin of error on user stories is also found to reduce the manifestation and effects of the bias.

5.2 Conclusions

Analysis of historic literature has shown that IT-project management is prone to planning fallacy bias and can affect results on time, costs, quality and functionality. A primary causing factor of the bias is the complex innovative character of IT projects with often little historical experience to base future planning on. Extant literature shows that the bias frequently manifests in traditional waterfall projects where planning is inflexible. This thesis studies the planning fallacy bias in agile-scrum IT projects. The bias is found to manifest in agile-scrum projects, yet the manifestation is limited compared to traditional IT projects as it is possible to intervene and adjust the planning as the project progresses. Iterative planning and optimization ceremonies appear to limit the extent to which the bias affects project results.

The risk of the planning fallacy bias manifesting in agile-scrum IT projects can be limited by ensuring that the project team and its members are experienced in using agile-scrum. Starting with rightly applying the refinement sessions to in detail understanding what needs to be done and finally, retrospective sessions to evaluate planning accuracy and take action to do better. In parallel it is key to ensure that decision power lies within the scrum-team and not with stakeholders. Providing transparency on the planned actions and impediments to the wider organization to build trust is seen as an enabler. All of these elements are part of the described agile manifesto.

Based on these findings it is concluded that the agile-scrum methodology can reduce the effects of the planning fallacy bias, yet the organizational maturity seems to present a challenge. The understanding of the importance of applying agile-scrum and ensuring it is applied correctly can aid to develop a sustainable adoption of the methodology and countermeasures.

5.3 Reflection

This subsection provides a reflection on the findings of this research in view of methodological and researcher limitations, which should be considered in assessing the findings.

Reflection on findings

The literature review has provided insight on the bias in the agile-scrum IT-context as it reveals specific appearance characteristics, causes, effects as well as countermeasures. The vast majority of the reviewed studies were experiments, where respondents were provided with fictional cases and tested on their forecasting accuracy (Buehler, Griffin, & Ross, 1994). The case context was rather general with the main aim to track the accuracy of planning and forecasting. One could argue that this type of situation is less realistic as there will be no actual consequences for respondents of their actions.

This study on the other hand has an exploratory character, using semi-structured interviews to understand the bias based on agile-scrum experience of the participating experts. The experts were specifically requested to answer all interview questions based on experience, using situation descriptions to elaborate. In this exploratory setup, that allows for follow-up questions and so a detailed data insight was obtained. Yet the setup might also affect the reliability as the interviewees are sharing past experiences that in no way can be confirmed.

Another important reflection made is the difference found in the effects of the planning fallacy bias. This study highlights the flexibility around which variables are fixed and which are not fixed. This is specifically interesting as the agile-scrum methodology prescribes the setup of costs and time as fixed variables. This phenomenon was not specifically noted in the literature review studies.

Reflection on methodology

The exploratory research setup is among the first studies to examine the planning fallacy bias in a real-life agile-scrum setting (Benschop, Rijsenbilt, & Wilmink, 2020). Biases often occur unconsciously yet the experience-based interview setup allows retrospective examination on if and how the bias manifested and affected a project. The importance of developing the interview questions based on the historical literature review is key to ensure the right questions are asked. Secondly, the limited sample size used in this study is adequate for discovering and suggesting new knowledge, but the results cannot be generalized as statistical testing or verification cannot be done. Finally, some specific remarks are made:

- The various organizations have different levels of experience or maturity while the interview results show that the experience level of the organization plays a key role in identifying if and to what extent the bias affects project results;
- Among interviewees the distribution in terms of different scrum roles was not very well balanced, partly due to time and agenda restrictions;
- The various interviewers may have differed in their approach to the semi-structured interview setup and the follow-up questions it allows;
- The interviews have been conducted and transcripts are made partly in Dutch, which poses the risk of interpretation of the researcher when translating and interpreting the Dutch interview data.

The above-mentioned elements may have had an influence on the findings of this study. In retrospect it would have been preferred to standardize the number of interviewees and their roles. Having more interviewees from specific roles participating may have impacted the results negatively and may emphasize specific elements more than others. To some extent this would have mitigated the differences in experience and maturity between organizations. The multi-interviewer method as well as the different organizations with interviewees speaking different languages should have been avoided to increase reliability overall and ensure that reduced reliability cannot be insinuated. A potential method to increase the internal validity will be shared in the future research recommendation section.

5.4 Implications

Implications for research

A primary contribution of this study is examining the role of the planning fallacy bias in an agile-scrum project setting. Specific elements of agile-scrum such as refinement and retrospective were described in detail and connected to the bias as well as the bias effects. As there is limited understanding yet on how the bias manifests in agile-scrum IT projects the findings of this study are revealing both for researchers developing- or validating theory on agile-scrum implementation as well as researchers who do so for the planning fallacy bias. Secondly, the exploratory interview setup will enhance triangulation as this research setup differs from most historic studies and so shines new light on the planning fallacy bias phenomenon from a different angle.

Implication for practice

This study provides organizations insights in what they can undertake to reduce the manifestation and effects of the planning fallacy bias. A first step toward optimization is to ensure that within an organization there is an aligned understanding, knowledge and, in time, experience on how to apply the methodology. Structural application of granular refinement and retrospective sessions to increase planning accuracy will raise awareness if unrealistic planning is at hand. This study found that the planning fallacy bias can result in an increased need of time and costs and in reduction of functionality and quality delivered. Prioritizing awareness of the bias and actively implementing and measuring the effectiveness of countermeasures in all agile-scrum project teams will have a real payoff.

5.5 Future research

To validate this supposition future research should focus on measuring the planning fallacy bias in experienced agile-scrum organizations. This type of research will allow a deeper validation of the found assumption in this thesis that the experience level of agile-scrum is a key requirement to minimize the manifestation of the bias. The further validation of the connection between the experience level and minimized effect of the bias can shine new light on the optimal application of agile scrum with minimized bias influence. Additionally, the setup of the research could be improved when it focusses on shared experiences among the participants. This may be done by focusing on a specific project and request the team members to participate in the exploratory semi-structured interviews. Using this setup will allow validation of shared experiences of one interviewee with those shared by another interviewee and so aid internal validity of the research.

References

Atkinson, R. (1999). Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. *International Journal of Project Management, 17*(6), 339-341.

Azanha, A. (2017). Agile project management with Scrum. *Emerald Insight, 10*(1), pp. 121-142.

Benschop, N., Rijsenbilt, A., & Wilmink, K. (2020). Cognitive Biases in Critical Decisions Facing SME Entrepreneurs: An External Accountants' Perspective. *Administrative Sciences*, 17-18.

Bjarnason, E. (2012). Are you biting off more than you can chew? A case study on causes and effects of overscoping in large-scale software engineering. *Information and Software Technology, 54*(10), pp. 1107-1124.

Bogdan, A., Cosmin, A., Sorin, C., & Costin, A. (2019). A study on using waterfall and agile methods in software project management . *JOURNAL OF INFORMATION SYSTEMS & OPERATIONS MANAGEMENT, 13*(1), pp. 129-131.

Brown, M. B., Dirska, H., & Pelosi, M. (2013, October). Agile Method Software Development Estimation Biases. *International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3*(Issue 10), 115-119.

Buehler et al. . (1994). Exploring the planning fallacy: why people underestimate their completion times. *Journal of Personality and Social Psychology - 51, 67*(3), pp. 1173 - 1182.

Buehler et al. . (2010). Finishing on time: When do predictions influence completion times? *Organizational and behavioural decision processes, 111*(1), 23-32.

Buehler et al. (1997). The role of motivated reasoning in optimistics time predictions. *Personality and Social Psychology Bulletin - 23, 23*(3), 238-247.

Buehler, R., Griffin, D., & Ross, M. (1994). Exploring the "Planning fallacy": why people underestimate their task completion times. *Journal of Personality and Social Psychology - 51*, 1173 - 1182.

Buhl, & Meier. (2011). The responsibility of business and information systems engineering in large scale IT projects. *Business & Information Systems Engineering, 3*(2), 61-64.

Buschmann, F. (2010). Unusable Software Is Useless. *IEEE Computer Society Digital Library Journals, 29*(1).

Cervone, H. (2011). Understanding agile project management methods using Scrum. *OCLC Systems & Services: International digital library perspectives, 27*(1), pp. 18 - 22.

Charette, R. (2010). Why Software Fails. *Institute of Electrical and Electronics Engineers, 42*(9), pp. 42 - 49.

Court, D., & Abbas, R. (2013). Whose Interview Is It, Anyway? *19*(6), pp. 480-488.

Drummond, H. (1996). Escalation in Decision-Making: The Tragedy of Taurus. *Oxford University Press, 173*(3), pp. 1139-1160.

Flyberg, B. (2006). Five Misunderstandings About Case-Study Researc. *Sage Publications, 12*(2), pp. 219-245.

Flyvbjerg, B. (2006). From Nobel Prize to project management: Getting risks right. *Project Management Journal, 37*(3), pp. 5-15.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Informations Systems Research, 20*(3), pp. 28 - 35.

Gartner Inc. (2019). *Driving Cost Optimization Across the Enterprise.* Retrieved from gartner.com: https://www.gartner.com/en/doc/3898566-driving-cost-optimization-across-the-enterprise-an-executive-perspective

Gobo. (2011). Glocalizing methodology? The encounter between local methodologies. *14*(6), pp. 417-437.

Goldfinch, S. (2009). Pessimism, Computer Failure, and Information Systems Development in the Public Sector,". *Public Administration Review, 67*(5), pp. 917-929.

Helweg-Larsen, M., & Shepperd, J. (2001). Do Moderators of the Optimistic Bias Affect Personal or Target Risk Estimates? *Personality and Social Psychology Review, 5*(1), pp. 74 - 95.

Jani, A. (2010). Escalation of commitment in troubled IT projects: Influence of project risk factors and self-efficacy on the perception of risk and the commitment to a failing project. *International Journal of Project Management, 29*(7), pp. 934 - 945.

Jensen, M., & Meckling, W. (1976). Theory of the Firm: Managerial Behavior, Agency Costs and Ownership Structure. *Journal of Financial Economics, 1*(1), pp. 305 - 360.

Jorgensen, M., & Grimstad, S. (2008). Avoiding Irrelevant and Misleading Information When Estimating Development Effort. *IEEE Computer Society Digital Library Journals, 25*(3), pp. 73 - 87.

Kahneman. (2012). *Thinking fast and slow.* London: Allen Lane., pp. 50-62

Kahneman et al. . (2003). Delussion of succes. *Harvard Business Review, 81*(7), pp. 1 - 11.

Kahneman, D. (1979). Intuitive Prediction: Biases and Corrective Procedures. *Studies in the Management Sciences: Forecasting*., pp 1 - 8

Kahneman, D. (1994). New challenges to the rationality assumption. *Journal of Institutional and Theoretical Economics, 150*(1), pp. 18 - 36.

Kahneman, D., & Lovallo , D. (1993). Timid Choices and Bold Forecasts: A Cognitive Perspective on Risk Taking. *Management Science, 39*(1), pp. 17 - 31.

Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decisions under risk. *Econometrica, 5*(1), pp. 313-327.

Kahneman, D., & Tvsersky, A. (1973). Judgement under uncertainty: heuristics and biases. *Office of Naval, 185*(4157), pp. 1124 - 1131.

Keil , e. (2007). Escalation: The Role of Problem Recognition and Cognitive Bias. *Decision Scienes, 38*(3), pp. 391 - 421.

Keil et al. . (2012). The Effect of an Initial Budget and Schedule Goal on Software Project Escalation. *Journal of management information systems, 29*(1), pp. 53 - 77.

Keil, M. (2000). Why software projects escalate: An empirical analysis and test of four theoretical models. *MIS Quarterly, 24*(4), pp. 631 - 664.

Keil, M. (2010). Is your project turning into a black hole? *California Management Review, 53*(1), pp. 6 - 31.

Keil, M., & Mahring, M. (2010). Is Your Project Turning into a Black Hole? *California Management Review, 53*(1), 6-8.

Kuhrmann et al. . (2017). Hybrid software and system development in practice: waterfall, scrum, and beyond. *39*(1), pp. 30 - 39 .

Kutsch, E. (2009). Performers, trackers, lemmings and the lost: Sustained false optimism in forecasting project outcomes — Evidence from a quasi-experiment. *International Journal of Project Management, 29*(8), pp. 1070 - 1081.

Kwak, Y. H. (2003). *Brief History Of Project Management.* K. A. Carayannis: Quorum Books., Chapter 2

McGrath, R. (2013). Transient Advantage. *Harvard Business Review, 42*(4), pp. 2 - 4.

Niederman, F. (1991). Information Systems Management Issues for the 1990s. *Management Information Systems Research Center, 15*(4), pp. 475 - 500.

Nouri et al. . (2012). The role of experience on techno-entrepreneurs' decision making biases. *Management Science Letters,, 2*(6), pp 1957 – 1964.

Pawar, R., & Mahajan, K. (2017). Benefits and Issues in Managing Project by PRINCE2 Methodology. *International Journal of Advanced Research in Computer Science and Software Engineering, 7*(1), pp. 190 - 192.

Pezzo et al. . (2005). On the distinction between yuppies and hippies: Individual differences in prediction biases for planning future tasks. *Personality and Individual Differences 41, 41*(7), pp. 1359 – 1371.

Pinchovski, B., & Fink, L. (2020). It is about time: Bias and its mitigation in time-saving decisions in software development projects. *International Journal of Project Management, 38*(2), pp. 100 - 101.

Project Management Institute. (2013). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide).* Newtiwb Square, Pennsylvania: Project Management Institue, Inc., pp. 233

Ronen, B., & Pass, S. (2007). *Focused Operations Management: Achieving More with Existing Resources.* New Jersey: John Wiley & Sons., pp. 73 - 121

Sandler et al. (2017). Assessing the Potential for Bias From Nonresponse to a Study Follow-up Interview: An Example From the Agricultural Health Study. *American Journal of Epidemiology, 186*(4), pp. 395 - 404.

Sanna, L., & Chang, E. (2005). The Hourglass Is Half Full or Half Empty: Temporal Framing and the Group Planning Fallacy. *Group Dynamics: Theory, Research, and Practice, 9*(3), pp. 173 - 188.

Sauer, R. (2007). Why develop open-source software? The role of non-pecuniary benefits, monetary rewards, and open-source licence type. *Oxford Review of Economic Policy, 23*(4), pp. 605 - 619.

Saunders et al. . (2015). What is theory? In M. Saunders, *Research Methods for Business Students* (pp. 47-50). New York: Pearson.

Schwaber, K. (2004). *Agile Project Management with Scrum.* Microsoft Press., pp. 144 - 167

Shalev, E. (2014). Optimism bias in managing IT project risks: a construal level theory perspective. *European Conference on Information Systems* (pp. 13). Tel Aviv: Association for Information Systems.

Shastri, Y. (2017). Understanding the Roles of the Manager in Agile Project Management. *Software Engineering Conference* (pp. 45 - 55). Jaipur: ISEC.

Shmueli, O. (2016). Can the outside-view approach improve planning decisions in software development projects? *Information Systems Journal , 26*(4), pp. 395 - 418.

Sing, M. (2013). Case Study Determining the Probability of Project Cost Overruns. *Journal of Construction and Engineering management, 139*(3), pp. 146 - 158.

Sleesman, D. (2012). Cleaning up the big muddy: a meta-analytic review of the determinants of escalation of commitment . *Academy of Management Journal, 55*(3), pp. 541 – 562.

Stanovich, K. (1999). The domain specificity and generality of belief bias: searching for a generalizable critical thinking skill. *Journal of Educational Psychology, 91*(3), pp. 497 - 510.

Trumper et al. . (2018). Heuristics and Biases in Project Management. *PM World Journal, 7*(1), pp. 1 - 3.

Usman, M., & Mendes, E. (2014). Effort Estimation in Agile Software Development: A Systematic Literature Review. *Conference of predictive models in software engineering* (pp. 82 - 91). Karlskrona: Blekinge Institute of Technology.

West, R., & Stanovich, K. (2001). Heuristic and analytic processing: Age trends and associations with cognitive ability and cognitive styles. *Journal of Experimental Child Psychology, 83*(1), pp. 26 - 52.

Whetten, D. (1989). What Constitutes a Theoretical Contribution? *Academy of Management, 14*(4), pp. 490 - 492.

Whyte, G. (1986). Escalating Commitment to a Course of Action: A Reinterpretation. *Academy of Management Review (11:2)*, pp. 311 - 321.

Yin, R. (2018). *Case Study Research and Applications: Design and Methods* (Vol. 14). London: Sage., pp 3 - 48

Appendix A. Semi-structured interview protocol

Agile-scrum role

1. What is your role in agile IT projects and how many years of experience do you have with these types of projects?

Planning fallacy bias

2. Do you recognize the bias in agile-scrum IT projects? Did you experience challenges due to over-optimistic forecasting?
3. How would you quantify the frequency in which bias manifests from 0% - 100%?
4. What are typical characteristics that can cause the bias? What agile-scrum characteristics increase the effect of the bias?
5. What could be the impact of the bias on budget, time functionality and quality? What typical agile-scrum characteristics increase the effect of the bias?
6. How would you quantify the effect of the bias on costs between 1(none) and 7(very much)?
7. How would you quantify the effect of the bias on time between 1(none) and 7(very much)?
8. How would you quantify the effect of the quality on time between 1(none) and 7(very much)?
9. How would you quantify the effect of the functionality on time between 1(none) and 7(very much)?
10. In the context of agile projects, what could you do reduce the effect of the bias?

Escalation of commitment

11. Do you recognize the bias in agile-scrum IT projects? Did you experience challenges due to over-optimistic forecasting?
12. How would you quantify the frequency in which bias manifests from 0% - 100%?
13. What are typical characteristics that can cause the bias? What agile-scrum characteristics increase the effect of the bias?
14. What could be the impact of the bias on budget, time functionality and quality? What typical agile-scrum characteristics increase the effect of the bias?
15. How would you quantify the effect of the bias on costs between 1(none) and 7(very much)?
16. How would you quantify the effect of the bias on time between 1(none) and 7(very much)?
17. How would you quantify the effect of the quality on time between 1(none) and 7(very much)?
18. How would you quantify the effect of the functionality on time between 1(none) and 7(very much)?
19. In the context of agile projects, what could you do reduce the effect of the bias?

Sunk cost

20. Do you recognize the bias in agile-scrum IT projects? Did you experience challenges due to over-optimistic forecasting?
21. How would you quantify the frequency in which bias manifests from 0% - 100%?
22. What are typical characteristics that can cause the bias? What agile-scrum characteristics increase the effect of the bias?
23. What could be the impact of the bias on budget, time functionality and quality? What typical agile-scrum characteristics increase the effect of the bias?
24. How would you quantify the effect of the bias on costs between 1(none) and 7(very much)?
25. How would you quantify the effect of the bias on time between 1(none) and 7(very much)?
26. How would you quantify the effect of the quality on time between 1(none) and 7(very much)?
27. How would you quantify the effect of the functionality on time between 1(none) and 7(very much)?
28. In the context of agile projects, what could you do reduce the effect of the bias?

Appendix B. Draft introduction text invitation

Dear ..,

This is the invitation to an interview focused on decision biases. A bias is a systematic error in thinking that occurs when individuals are processing and interpreting information. Decision biases can affect decisions and judgements. With participating in this interview, you contribute to broad research on understanding if and how decision biases manifest in agile-scrum IT projects. As you are a knowledge matter expert on applying this IT-project management method your insights and experiences on whether or not these biases manifest is a great interest for the scientific development of understanding decision biases in this environment.

In an interview setting, you will be requested to answer questions related to whether or not the bias manifests in your project experience, what caused the bias and what was the consequence of the bias. The below-explained biases will be subsequently discussed:

1. **Planning fallacy bias**
   Definition: the systematic underestimation of the time required to complete activities. This can lead to unrealistic estimates of time, resources, costs & results.
   Practical example: the construction of the "Noord-Zuid Metro" in the Netherlands where the project was estimated at 1.6 billion euro and should take 5 years yet in practice it costed 3.1 billion euro and took 8 years to complete.
2. **Sunk cost bias**
   Definition: occurs when decision maker(s) are influenced by past investments in a projects. As a result, people feel compelled to continue with the projects and keep on investing in the project as the initial investment is already so high, so it is hard to pull the plug and take the loss.
   Practical example: a big project ran for a lot of years in your company but actually there is little progress. Though as there are millions spent it is impossible to stop – after all the money is invested for a reason so we have to continue in order to succeed. Individuals miss the question is the value actually still feasible to be delivered?
3. **Escalation of commitment**
   Definition: the tendency of decision makers to continue with failing projects. Decision makers remain too committed and continue to invest in resources even though it is no longer rational.
   Practical example: you have invested so much in a project in terms of resources, time and are so committed and believing that it will work that in practice you cannot stop any more – the commitment is too big to let go.

The interview will take up to 1.5 hours, and during the meeting, you will be taken through discussing all the biases.
Your time and support are much appreciated to increase understanding and prevent the harmful effects of the bias by taking countermeasures.

Thank you in advance,

With kind regards,

Appendix C. Description of codes

Description of the emerged interview themes and the related codes, the description and frequency of appearance of the codes in the eighteen interviews.

THEME	CODE	CODE DESCRIPTION	Frequency
Manifestation	Agile-scrum experience	This code is assigned to text fragments where the knowledge matter expert expresses that the extent to which the planning fallacy bias manifests depends on the team maturity in applying agile-scrum methodology. This code **is not** assigned when the manifestation is related to the IT technical experience.	13
	Intervene on unrealistic planning	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias manifests during the project, but the manifestation is limited due to the short iterations used in agile-scrum methodology and ability to adjust the planning. This code **is** assigned when both a limited manifestation, the short iterative character and ability to adjust planning are mentioned. This code **is not** assigned when only a reduced manifestation is mentioned.	9
	Scrum planning unrealistic	This code is assigned to text fragments where the knowledge matter expert expresses to recognize the planning fallacy bias in agile-scrum IT projects and describes the manifestation by the frequently identified unrealistic planning. This code **is not** assigned when the knowledge matter expert related unrealistic planning to traditional IT projects.	8
Causes	Overly optimistic	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias manifestation is caused by the human tendency to underestimate time needed to complete activities in IT projects.	14
	No proper agile-scrum structure	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias is caused by not applying the agile-scrum methodology but only applying parts of the methodology.	1
	No reflection/retrospective iterations	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias is caused by a lack of evaluative iterations and optimization actions based on the past activities and team collaboration.	5
	No clarity on what end-customer expects	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias is caused by an inaccurate understanding of the expectations of the end-customer and so a limited refinement of the user stories. This code **is not** assigned when lack of technical insights is mentioned.	3
	In-transparency of technical impediments	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias is caused by in-transparency of the technical impediments that might occur during the project. This code **is not** assigned when unclarity around end-customer expectations are mentioned.	9

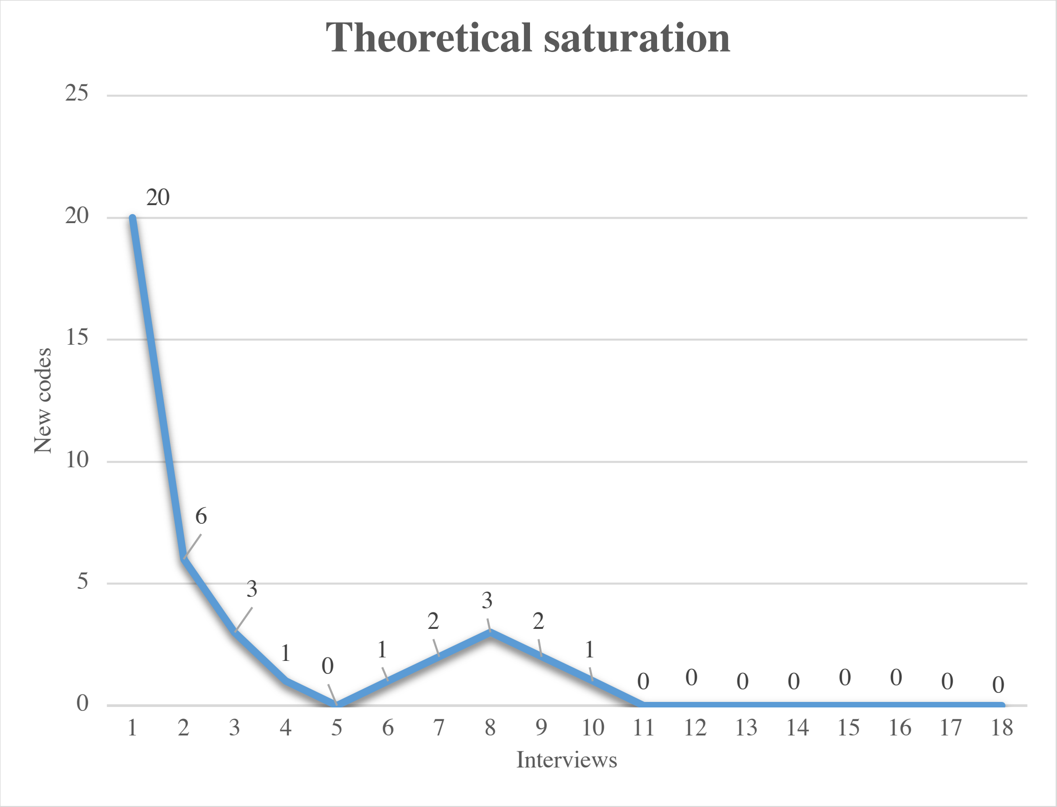
	Agile-scrum misconception	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias is caused by an inaccurate understanding of agile-scrum methodology and so unrealistic expectations of the applying the methodology.	5
	Limited scrum agile experience	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias is caused by the limited experience or limited maturity of the scrum-team in applying the agile-scrum methodology in IT projects.	9
	Complexity of IT Infrastucture	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias is caused by the complexity of the IT infrastructure with a low of unknown variables and interdependencies. This makes it hard to accurately plan and forecast activities.	12
	Affraid to dissapoint stakeholders	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias is caused by the fear of team members to disappoint or disagree with management or senior stakeholders. Which leads to agreeing deliverables and project planning that is actually not feasible. This code **is not** assigned when external pressure leads to for example reduced quality.	7
	Influenced by strong opinions of team members	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias is caused by dynamics in the team where team members can be influenced by other team members who have a strong opinion and disregard their challenges. This code **is not** assigned when team members are influenced by external factors.	2
Effects *time*	Delayed delivery	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias effects time by a completion of deliverables beyond the agreed deadline or agreed planning.	13
	Increased work hours towards delivery	This code is assigned to text fragments where the knowledge matter expert expresses that the effects of the planning fallacy bias lead to increased work hours of the existing scrum-team towards the delivery date to ensure things can be completed. This code **is not** assigned when references are made to adding additional people to the team.	5
	No delayed delivery	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias does not affect time as in the agile-scrum way of working time is a fixed variable. Functionality will be compromised over time.	11
Effects *costs*	Delayed delivery increases costs	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias effects costs as due to the inaccurate planning and delayed delivery the scrum-team needs to be utilized longer than anticipated at the start.	15
	Increase team size	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias effects leads to an increase of team capacity towards the delivery date or deadline as more people are needed than anticipated to finish the deliverables on time.	3

	Project costs not exceeded	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias does not affect costs as in the agile-scrum way of working costs are a fixed variable. Functionality will be compromised over costs.	11
Effects *Quality*	Quality compromised to meet deadline	This code is assigned to text fragments where the knowledge matter expert expresses that quality is compromised rather than functionality, time or costs in order to meet the delivery date or deadline.	9
	Increased re-work after delivery	This code is assigned to text fragments where the knowledge matter expert expresses that compromising quality leads to increased reported incidents and re-work activities after the delivery date.	4
	Quality is never compromised	This code is assigned to text fragments where the knowledge matter expert expresses that quality is never compromised as it is a fixed variable in the agile-scrum way of working. Time, costs or functionality will be compromised over quality.	11
Effects *Functionality*	Functionality delivery delayed	This code is assigned to text fragments where the knowledge matter expert expresses that the planning fallacy bias effects functionality by a delayed delivery of functionality / the minimum viable product beyond the agreed deadline or agreed planning.	13
	Dissapointing end-customer and stakeholders	This code is assigned to text fragments where the knowledge matter expert expresses that the effects of the planning fallacy bias lead to often disappointing stakeholders or the end-customer who were counting on having functionality available at an agreed moment.	8
	Functionality is never compromised	This code is assigned to text fragments where the knowledge matter expert expresses that functionality is never compromised as it is a fixed variable in the agile-scrum way of working. Time or costs will be compromised over functionality.	5
Countermeasures	Plan slack	This code is assigned to text fragments where the knowledge matter expert expresses that the effect of the planning fallacy bias can be reduced by structurally planning extra time to complete activities.	3
	Refined and detailed user stories	This code is assigned to text fragments where the knowledge matter expert expresses that the effect of the planning fallacy bias can be reduced by planning as granular as possible with input from the end customer.	13
	Definition of done	This code is assigned to text fragments where the knowledge matter expert expresses that the effect of the planning fallacy bias can be reduced by having an aligned understanding of the desired output.	4
	Learn from experience	This code is assigned to text fragments where the knowledge matter expert expresses that the effect of the planning fallacy bias can be reduced by structurally evaluating what went well and what should be improved and then actively implementing and measuring optimizations.	11
	Agile-scrum experience	This code is assigned to text fragments where the knowledge matter expert expresses that the effect of the planning fallacy bias can be reduced by a sufficient experience level of the team to apply the agile-scrum methodology and actively increasing this maturity level by training and coaching.	4

	Team size and dedicated availability	This code is assigned to text fragments where the knowledge matter expert expresses that the effect of the planning fallacy bias can be reduced by ensuring a team size that is sufficient to complete the deliverables and ensuring that the team works dedicated on a project.	1
	Stakeholder- and Management support	This code is assigned to text fragments where the knowledge matter expert expresses that the effect of the planning fallacy bias can be reduced by management support and trust to allow the team to make mistakes, fail and learn.	6
	Transparency in team and towards management and stakeholders	This code is assigned to text fragments where the knowledge matter expert expresses that the effect of the planning fallacy bias can be reduced by providing structural transparency in the team and to stakeholders on what the development status is, what impediments are identified and mitigated, what will be delivered when and what activities are planned.	4
	Planning poker	This code is assigned to text fragments where the knowledge matter expert expresses that the effect of the planning fallacy bias can be reduced or mitigated by structurally embedding "planning poker" sessions to prioritize activities with the team.	4
	Visualize interdependencies	This code is assigned to text fragments where the knowledge matter expert expresses that the effect of the planning fallacy bias can be reduced or mitigated by to the team and other scrum teams visualize the identified interdependencies.	2
	Learning among scrum teams	This code is assigned to text fragments where the knowledge matter expert expresses that the effect of the planning fallacy bias can be reduced or mitigated by structurally providing sessions where scrum teams share learnings.	2
	Experienced scrum-master	This code is assigned to text fragments where the knowledge matter expert expresses that the effect of the planning fallacy bias can be reduced or mitigated by ensuring an experienced scrum-master leads the team and facilitates the team to increase planning accuracy.	3

Appendix D. Interview saturation

Visual representation of the theoretical saturation shown by the new codes identified per interview. In this overview the new codes of all themes are combined.


Theoretical saturation
New codes
Interviews