With agile software development approaches comes an increased dependability on well-functioning teams that are profoundly influenced by social-psychological factors.

The objective of this research was to investigate if and how psychological group processes, i.e. the temporal perspective often referred to as group development, are related to what is meant by an agile team.

A diversity of research designs and data collection methods was used, including surveys, interviews, and project data, to find and explain connections between team agility and group developmental stages. A total of 311 people participated in the studies from 19 different companies situated in the US, Brazil, The Netherlands, and Sweden.

The results show that there are connections between group development maturity and what is meant by an agile team, demonstrating the relevance of psychological group processes when building agile teams. Group developmental issues were related to many aspects of how team agility is described, including team planning effectiveness, interpersonal conflict, open communication, and dedication.

We conclude that agile teams at different group development stages adopt team agility differently, and the implementation and management of agile projects need to be adapted to what stage the team is in from a group-developmental perspective.

Lucas Gren
Division of Software Engineering
Department of Computer Science and Engineering

Lucas Gren Psychological Group Processes when Building Agile Software Development Teams

2017

ISBN 978-91-982237-3-6

UNIVERSITY OF GOTHENBURG

Psychological Group Processes when Building Agile Software Development Teams

Lucas Gren

Ph.D. thesis

Department of Computer Science and Engineering
University of Gothenburg, Sweden 2017

IT Faculty

Thesis for The Degree of Doctor of Philosophy (Ph.D.)

Psychological Group Processes when Building Agile Software Development Teams

Lucas Gren

Division of Software Engineering
Department of Computer Science and Engineering
Chalmers University of Technology and
The University of Gothenburg
Gothenburg, Sweden, 2017

Psychological Group Processes when Building Agile Software Development Teams

Lucas Gren

Supervisor: Professor Richard Torkar
Co-supervisor: Associate Professor Richard Berntsson Svensson
Examiner: Professor Michel Chaudron

Technical Report No 147D
ISBN 978-91-982237-3-6
Department of Computer Science and Engineering
Division of Software Engineering
Chalmers University of Technology and
The University of Gothenburg
Gothenburg, Sweden

e-publication: http://hdl.handle.net/2077/54190

This thesis was typeset with LaTeX.
Printed by Chalmers Reproservice,
Gothenburg, Sweden 2017.

"Cognitive psychology tells us that the unaided human mind is vulnerable to many fallacies and illusions because of its reliance on its memory for vivid anecdotes rather than systematic statistics." — Steven Pinker

Abstract

Background With the agile approach to managing software development projects, comes an increased dependability on well functioning teams. Agile teams are profoundly influenced by social-psychological factors since more communication and cooperation are needed both within the organization and team, but also with customers.

Objective The objective of this thesis is to investigate if and how psychological group processes, i.e. the temporal perspective often referred to as group development, is related to what is meant by an agile team.

Method A diversity of research designs and data collection methods were used, including surveys, interviews, and project data, to find and explain connections between team agility and group developmental stages, but also agile maturity model validity and individual nontechnical skills. A total of 311 people participated in the studies from 19 different companies situated in the US, Brazil, The Netherlands, and Sweden.

Results The results show that there are connections between group development maturity and what is meant by an agile team, demonstrating the relevance of psychological group processes when building agile teams. Group developmental issues were related to many aspects of how team agility is described, including team planning effectiveness, interpersonal conflict, open communication, and dedication. Moreover, the mature use of agile practices could not be explained by individual nontechnical skills and the efficiency of task implementation in agile software development teams were not dependent of group maturity, but instead individual technical skills.

Conclusions Our first conclusion is that many agile measurement scales are not scientifically validated and the construct of *agility* needs to be broken down into parts that need to be researched separately; one such part being what is meant by *team agility*. Secondly, agile teams at different group development stages adopt team agility differently, and the implementation and management of agile projects need to be adapted to what stage the team is in from a group-developmental perspective. We also conclude that efficiency, but not effectiveness, in agile software development might be more dependent on individual technical skills than group development and that individual nontechnical skills are poor predictors of the maturity of agile practices.

Keywords agile teams, group development, social psychology, software engineering

Technical Report No 147D
ISBN 978-91-982237-3-6

Acknowledgment

I would first like to acknowledge all the people I have been in contact with at the participating companies. Without you believing in my research these scientific publications would not have existed. I would like to thank my supervisor Professor Richard Torkar, my co-supervisor Associate Professor Richard Berntsson Svensson, my examiner Professor Michel Chaudron, Professor Robert Feldt, Professor Alfredo Goldman, Associate Professor Christian Jacobsson, and the former Master's students Khaled Al-Sabbagh and Konstantinos Chronis. I would also like to thank the most perfect person I have ever met, that is, my daughter Linn, for being a part of my life, my love Maja, my family and especially JO (R.I.P.), my colleagues, and all my friends who are all very loving and supportive.

List of Publications

Appended papers

This thesis is based on the following papers:

1. L. Gren, R. Torkar, R. Feldt "The prospects of a quantitative measurement of agility: A validation study on an agile maturity model"
The Journal of Systems and Software, 107, 38–49, 2015.

2. L. Gren, R. Torkar, R. Feldt "Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies"
The Journal of Systems and Software, 124, 104–119, 2017.

3. L. Gren "The links between agile practices, interpersonal conflict, and perceived productivity"
Published in the Proceedings of the 21st Conference on Evaluation and Assessment in Software Engineering (EASE), Karlskrona, Sweden, June 15–16, 2017 pp. 292–297.

4. K. Al-Sabbagh, L. Gren "The connections between group maturity, software development velocity and planning effectiveness"
The Journal of Software: Evolution and Process (Accepted August 8, 2017).

5. L. Gren, A. Knauss, C. Johann Stettina "Non-technical individual skills are weakly connected to the maturity of agile practices"
In submission to journal.

Papers not included in the thesis

The following papers are not included in the thesis:

1. L. Gren, R. Torkar, R. Feldt "Work motivational challenges regarding the interface between agile teams and a non-agile surrounding organization: A case study"
Published in the Proceedings of Agile Conference (AGILE), Orlando, Florida, July 28–August 1, 2014 pp. 11–15.

2. L. Gren, R. Torkar, R. Feldt "Group maturity and agility, are they connected? – A survey study"

Published in the Proceedings of the 41st Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Madeira, Portugal, August 26–28, 2015 pp. 1–8.

3. R. Berntsson Svensson, M. Taghavianfar, L. Gren "Creativity techniques for more creative requirements: Theory vs. practice"
Published in the Proceedings of the 41st Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Madeira, Portugal, August 26–28, 2015 pp. 104–111.

4. L. Gren, A. Goldman "Useful statistical methods for human factors research in software engineering: A discussion on validation with quantitative data"
Published in the Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), Texas, USA, May 16, 2016 pp. 121–124.

5. K. Chronis, L. Gren "Agility measurements mismatch: A validation study on three agile team assessments in software engineering"
Published in the Proceedings of the 17th International Conference on Agile Software Development (XP), Edinburgh, United Kingdom, May 24–27, 2016 pp. 16–27.

6. L. Gren "Using the agile adoption framework to assess agility and guide improvements"
Published in the Proceedings of the Work in Progress session held in connection with the 42nd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Limassol, Cyprus, August 31–September 2, 2016, ISBN 978-3-902457-46-2, pp. 1–2.

7. L. Gren "The systems approach to change and the agile software development context"
Published in the Proceedings of the Psychology of Programming Interest Group Workshop (PPIG), Cambridge, UK, September 7–9, 2016 pp. 1–4.

8. L. Gren, A. Goldman "Trying to increase the mature use of agile practices by group development psychology training — An experiment"
Published in the Proceedings of the 4th International Workshop on Quantitative Approaches to Software Quality (QuASoQ), 2016, Hamilton, New Zealand, December 6, 2016 pp. 50–57.

9. L. Gren, K. Al-Sabbagh "Group developmental psychology and software development performance"
Published in the Poster Track Proceedings of the 39th International Conference on Software Engineering Companion (ICSE-C), Buenos Aires, Argentina, May 24, 2017 pp. 232–234.

10. L. Gren, R. Berntsson Svensson, M. Unterkalmsteiner "Is it possible to disregard obsolete requirements? — An initial experiment on a potentially new bias in software effort estimation"

Published in the Proceedings of the 10th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), Buenos Aires, Argentina, May 23, 2017 pp. 56–61.

11. L. Gren "A fourth explanation to Brooks' Law — The aspect of group developmental psychology"
Published in the Proceedings of the 10th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), Buenos Aires, Argentina, May 23, 2017 pp. 86–89.

12. L. Gren, V. Antinyan, "On the relation between unit testing and code quality"
Published in the 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Vienna, Austria, August 30–September 1, 2017 pp. 52–56.

13. L. Gren "Social influence in agile requirements engineering"
Published in the Proceedings of the Work in Progress session held in connection with the 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Vienna, Austria, August 30–September 1, 2017 pp. 2–4.

Contents

Chapter 1

Introduction

In the context of software engineering, a set of practices is applied to manage complex and rapidly changing projects. These practices have become extremely popular both within (see e.g. [1]), and outside software development (see e.g. [2, 3]). In contrast to traditionally plan-driven software development, this paradigm was given the name *agile*, which means to be able to move quickly, easily and be flexible [4]. In complex environments, where clear goals are hard to define and ever-changing, flexible managerial styles have been suggested to deliver customer value also in earlier management research (see e.g. [5]), which is expressed as *working software* in agile software development [6]. Working software should be delivered at an early stage and is then continuously developed in close customer collaboration, rather than being subject to the uncertainties of large up-front design and planning in a changing environment [6]. According to Cobb [7], agile principles are applicable when the project has demands that are hard to define, or when the client does not know what (s)he exactly wants. The principles are less suitable in situations where these conditions do not apply, i.e. projects with a firm time-frame, clear goals, and with little to no uncertainties so that early and up-front specification can be used [7]. The multitude of reported benefits of an agile approach includes increased customer collaboration, better estimation of tasks, and increased quality [8], but also higher job satisfaction [9] as well as overall stakeholder satisfaction and therefore project success [10]. Agile development, compared to plan-driven ditto, implies more communication and a stronger focus on people, which make the social-psychological aspects critical to understand [11].

A few studies have been conducted that set out to investigate some social-psychological aspects of agile development. Whitworth et al. [12], for example, verify that agile teams need to look at social-psychological aspects to fully understand how they function. There are also studies connecting agile methods to organizational culture [13, 14]. These connect the agile adoption process to organizational culture to see if there are cultural factors that could jeopardize the agile implementation, which there are. One study divides corporate culture into different layers according to Schein [15]. That article shows that an understanding of cultural layers increases the knowledge of how an agile organizational culture could be established [16].

A more recent study has underlined the importance of focusing even more on social-psychological aspects of workgroups (or teams) in software engineering to gain more descriptive and predictive power [17].

It is important to realize that other research fields had conducted extensive research on teams long before software engineering existed. Therefore, this thesis includes an overview and an evaluation of agile measurement models that point to the issues of measuring a new and somewhat ill-defined construct (Paper 1). Paper 1 made us realize that there is a set of psychological aspects not included in the measurements as well as the fact that very few of the measurements are scientifically validated. From a team perspective, measurement is essential to evaluate progress [18]. However, one of the most significant challenges is to measure at the right abstraction level [19]. Hackman [19] divides levels of abstraction into three, namely micro, meso, and macro. When researching the creation of agile teams, we want to focus on group-level development, hence, the meso level. However, explanations to variance we observe might very well come from the macro (organizational) level or the micro (individual) level. Therefore, if we want to understand agility and how such cultural change affects a software development organization, we should, preferably, look at all three levels. Paper 2 includes both the meso and macro levels, and one part is a high-level investigation of how the people responsible for the 'agile process' reason around the enterprise agility in connection to their specific software development teams. However, the quantitative data were collected concerning the group-level of the agile teams. Paper 3 also focuses on the team-level and is a correlation analysis of interpersonal team conflict in relation to the maturity of agile practices used and the perceived team productivity. Paper 4 further investigates the connection between group (or team) maturity regarding external measurements of productivity, namely that of software development velocity and planning effectiveness. To see if variance could be explained on the micro level, Paper 5 investigates if individual non-technical skills could explain the maturity of agile practices, which they could not.

Some studies try to connect psychological aspects to software engineering teams in general, which are described in Section 1.2. For the reader to fully understand the content of this thesis, a basic understanding of agile software development and its underlying principles is needed, and therefore, Section 1.2 also includes an overview of agile software development in connection with existing research in social psychology, management, and organizational psychology. When it comes to measuring agility, a background to agility research concerning agile maturity models can be found in the Related Work section in Paper 1 (Chapter 2). Also, the reader needs to comprehend basic group developmental psychology, which is also briefly introduced in Section 1.2 while a more comprehensive description is given in the Related Work section of Paper 2 (Chapter 3). An additional explanation of group developmental psychology is also provided in the Background Section of Paper 4 (Chapter 5), together with a description of existing software process improvement models.

In the next section (Section 1.1), the research focus for this thesis is presented including the research goal and the research questions. After that, Section 1.2, will present the theory that is not included in the theoretical parts of the appended papers, including detailed descriptions of agile soft-

ware development from a psychology perspective, and more general related work to that of psychological aspects of software engineering. Section 1.3 will first present a philosophical reflection on scientific discovery and then show what methods that were used in the appended papers. Section 1.4 presents a summary of each paper and their contribution. Section 1.5 will discuss the papers' contributions in connection the theoretical background to answer the overall research question and present limitations to this thesis. Section 1.6 is an analysis of the validity threats of each paper and in conjunction. Finally Section 1.7 will provide conclusions and suggest future work.

1.1 Research focus

The high-level goal of this thesis is to look more closely at what team agility is from a group-psychological perspective. We make no claim having defined 'agility' in general, only adding the temporal perspective of team dynamics to what is meant by an 'agile team.' As illustrated in Figure 1.1, we started out by reviewing how other researchers propose we measure agility in the software engineering domain. The conclusions from that study (Paper 1) was that we need to apply more rigorous scientific methods to even state that we measure parts of behavior that could be seen more agile than the traditional management approaches used before.

We had the hypothesis that agility in the software engineering context has a lot to do with group dynamics and relationships between people in the development life-cycle, and more so than plan-driven development. To test this hypothesis, we investigated the effects of group maturity on agile practices adoption. We conducted an interview study (Paper 2) together with data from our newly created factors from Paper 1. This analysis shined a light on how group maturity affects the agile practices and how people responsible for the agile processes work to navigate through these psychological hurdles when building agile teams. In Paper 3, we focused on the conflict stage and investigated the hypothesized negative relationship between interpersonal conflict in teams and their mature use of agile practices. The fourth step (Paper 4) was to look at external performance measures to collect further evidence of the effect of different group maturity stages on agile practices. Paper 5 is an investigation of the individual level to see if individual nontechnical skills could predict the maturity of agile practices, which they could not. Such a result points at the appropriateness of looking at team agility using the team-level as the unit of analysis.

The research questions from the appended papers helped us find an answer to the main research question:

- RQ: How is group maturity related to team agility in software development?
  - RQ1: Is the agile adoption framework valid according to quantitative tests for internal consistency and construct validity? (Paper 1)
  - RQ2: How is group maturity connected to building agile teams? (Paper 2)

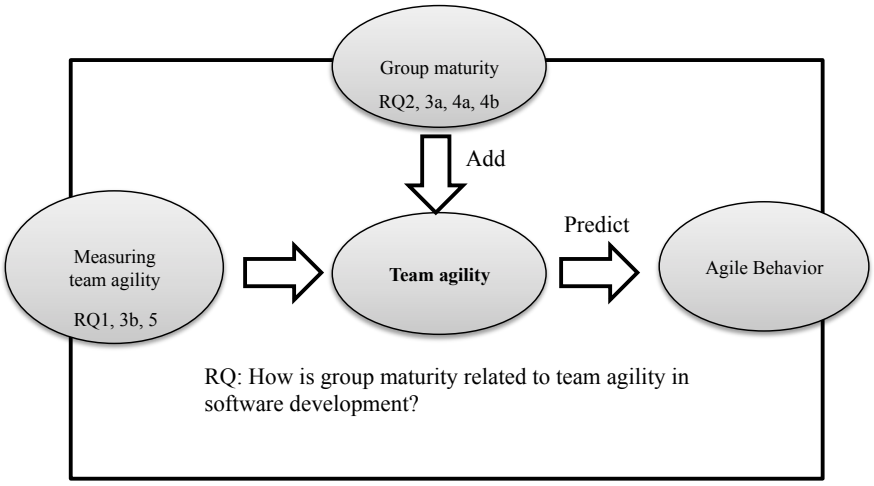

Figure 1.1: Research focus.

- RQ3a: Which, if any, agile practices are positively or negatively associated with interpersonal conflict? (Paper 3)
- RQ3b: Which, if any, agile practices are positively or negatively associated with perceived productivity? (Paper 3)
- RQ4a: What is the association between group maturity and planning effectiveness? (Paper 4)
- RQ4b: What is the association between group maturity and software development velocity? (Paper 4)
- RQ5: Are individual nontechnical skills connected to the mature use of agile practices? (Paper 5)

We will now give a short overview of agile software development research and analyze the agile work practices from a psychological perspective taken mostly from social psychology research on groups.

1.2 Background and related work

1.2.1 Agile software development

The basic idea of agile software development is that complex projects need to combine the traditional approach to managing projects and the need to be able to respond to change. A core practice of any agile method is to develop the product iteratively, meaning that the projects are divided into shorter iterations (sometimes called 'sprints') so that the requirements (sometimes called 'user stories') can be re-prioritized continuously. Compared to other management approaches, this is the key difference in agile software development, since it is the only project management method that assumes the end-goal is unknown in detail throughout the project life-cycle [20]. Another key idea about how such responsiveness to change should be possible is to focus more on social-psychological aspects both within the teams and with customers [21]. The agile community has, thus, defined a set of principles that they summarize in the Agile Manifesto [4]: "We are uncovering better ways of developing

software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

That is, while there is value in the items on the right, we value the items on the left more." [4].

In an attempt of making the manifesto more concrete, the authors connected a set of twelve principles to their manifesto that has been reviewed by Williams [4]:

1. Our highest priority is to satisfy the customer through the early and continuous delivery of valuable software.
2. Welcome changing requirements at the start of each iteration, even late in development; agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter time-scale.
4. The whole team, from business people through testers, must communicate and collaboratively work together throughout the project.
5. Build projects around empowered, motivated individuals with a shared vision of success; give them the environment and support they need, clear their external obstacles, and trust them to get the job done.
6. The most efficient, effective method of conveying information to and within a development team is through synchronous communication; important decisions are documented so [they] are not forgotten.
7. Valuable, high-quality software is the primary measure of progress at the end of each short time-boxed iteration.
8. Agile processes promote sustainable development. The whole team should be able to maintain a reasonable work pace that includes dedicated time for exploration, visioning, refactoring, and obtaining and responding to feedback.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity — the art of maximizing the amount of work not done — is essential.
11. The best architectures, requirements, and designs emerge from self organizing teams guided by a vision for product release.

12. With each iteration, the team candidly reflects on the success of the project, feedback, and how to be more effective, then tunes and adjusts its plans and behavior accordingly.

From both a research and industry perspective, measuring 'agility' is of course of very high value. However, the vagueness and breadth of the agile principles make a definition of one single construct of 'agility' in its broader sense complicated to achieve. The research on agile software development has, therefore, naturally become split into different subcategories, like what is meant by 'agile requirements engineering' [22], 'agile contracting' (see e.g. [23]), and so on and so forth.

1.2.2 Research on agile software development

There are many agile methods, such as eXtreme Programming (or XP) [20], Kanban [24] and Scrum [25], which try implement the agile principles by prescribing a set of concrete practices. Even though these practices are widely used in industry [26], their scientific underpinnings can sometimes be questioned. Much of the literature is from anecdotal evidence presented by practitioners in books with little available data (see e.g. [7,27,28,29,30,31,32,33,34]). However, in recent years, the empirical studies on agile software development have increased and focus on different aspects of agile software development and management [35]. The areas covered by secondary studies were: adoption, methods, practices, human and social aspects, CMMI, usability, global software engineering (GSE), organizational agility, embedded systems, and software product line engineering [35]. The three systematic literature reviews covering human and social aspects and organizational agility were about dimensions of organizational agility (organizational structures, workforce, development process, management and leadership, and infrastructure) [36], the role of communication [37], and developers' motivation in agile projects [38]. In a more recent secondary study on behavioral software engineering (i.e. human and social aspects of software engineering), Lenberg et al. [17] concluded that studies have mostly: "focused on a few concepts, which have been applied to a limited number of software engineering areas." Furthermore, they conclude that "the individual studies have typically had a narrow perspective focusing on few concepts from a single unit of analysis." Wufka et al. [39] also concluded that agility emerges from teams' reactions to needs for change and that agility needs to be understood from a process (i.e. temporal) perspective of teams.

Altogether recent research on agile software development defines a gap with regards to using many different units of analysis, and no studies have been found that specifically connect the temporal perspective of group dynamics (i.e. group development) and agile software development teams.

We have only found one study that has summarized the practices that are common to most agile methods created a scale for each practice and validated these factors by using psychometrics [40]. We used these scales as an agile practices measurement in Papers 3 and 5 that are included in this thesis. One of the authors, Chaehan So [41], also presents a unique quantitative study on agile practices in connection with some aspects of social psychology. His study provides an empirically evaluated connection between these agile practices and measurements of goal commitment, social support, adaptation, and knowledge

growth (even though in the form of a non-peer-reviewed dissertation in social psychology). The author also suggests a useful division of the agile practices into *core*, *technical*, *team interaction*, and *customer interaction* practices.

In the next section, we will present more research on agile software development in connection with these agile work practices and to management, social and organizational psychology findings.

1.2.3 Agile work practices and social psychology

We will now describe the general agile work practices, as defined by So et al. [40], and connect these to existing social, management, and organizational psychology findings. For a reader without a background in social psychology to understand some of the reflections we will first provide a set of definitions:

- Group – "Three or more members that interact with each other to perform a number of tasks and achieve a set of common goals" [42].
- In-group – "A group that an individual is a member of" [43].
- Out-group – "A group that an individual is not a member of" [43].
- Entitativity – "The property of a group that makes it seem like a coherent, distinct and unitary entity" [43].
- Group socialization – "Dynamic relationship between the group and its members that describes the passage of members through a group in terms of commitment and of changing roles" [43].
- Group structure – "Division of a group into different roles that often differ with respect to status and prestige" [43].
- Roles – "Patterns of behavior that distinguish between different activities within the group, and that interrelate to one another for the greater good of the group" [43].
- Stereotype – "Widely shared and simplified evaluative image of a social group and its members" [43].
- Prototype – "Cognitive representation of the typical/ideal defining features of a category." A prototype is only a stereotype if shared by the out-group [43].
- Self-esteem – "Feelings about and evaluations of oneself" [43].

Iterative development – The core practice of agile development The agile principles one, three, and seven (see Section 1.2.1) are all directed towards delivering value in short iterations to customers, highlighting this practice as the core practice of agile software development [44]. The idea is to deliver working software comprising some initial functionality that gives value to the customer at a very early point in time in the development life-cycle. Such approaches have high face validity, but when broken down, these ideas include a diversity of competences and dynamics needed by the agile team to

deliver value in such short iterations. In more general management research, there has been more thorough research on which general work practices contribute to performance (see e.g. [45]) and to successfully implement iterative development, the team must have a high degree and maturity of, for example, staffing, decentralized decision-making, and communication [46]. Principle two is about welcoming changes in requirements from customers and integrate these changes into the upcoming delivery. Such responsiveness to change demands decentralized decision-making, but also self-organization (or self-management) of teams within the organization [46]. Without the possibility to self-organize, the structure will be too rigid to enable the team to respond as rapidly as necessary in the software development business [21], an aspect also stated in principle number eleven.

Iteration planning – A teamwork practice Agile principle number five spans over a broad area of psychological aspects of the workplace. Obtaining empowered and motivated individuals that have the needed support to solve any given task together with high levels of trust, are all aspects known to be necessary [47,48] but are not always in place [49]. Creating a shared vision has also been shown in research to be a key for success since the beginning of the 1990s and is one of the main components of transformational leadership [50]. A shared vision is necessary since the team needs an overall goal to break down when planning the upcoming iteration. The process of planning an iteration is sometimes called the *Planning Game* in the agile context [51], where the team (including the customer or a customer representative) prioritizes and conducts effort estimation on the requirements that are often written as 'user stories' [52]. Principle number ten regarding the importance of simplicity is somewhat connected to the concept of reducing waste in lean manufacturing, together with the continued avoidance of doing unnecessary activities in the project (or process) life-cycle [53]. To plan in such a way, the team must know the members' real competences and abilities, which also implies maturity in the development process and that the members of the group are committed and fully integrated into the group (see Levine et al. [54] for more details on group socialization theory).

Stand-up meetings – A teamwork practice The fifth principle states that developers, but also business people and testers, should be on the same team and collaboratively work together through the whole project life-cycle (i.e. having cross-functional teams). For teams to have the possibility to coordinate, the agile teams have frequent team coordination meetings that also need to be time-boxed (a rule of thumb is around 15 minutes which is the amount of time people can 'stand up'). The three most common questions all group-members are intended to answer on a daily basis are: (1) What have you done since we last met? (2) What are you planning to do until we meet again? (3) What, if any, impediments are you encountering that are preventing you from making progress? [55].

A well-researched approach to explaining many group phenomena is the social identity theory (see e.g. [43, 56]). Not only has the theory gained empirical evidence in social psychology research but also in recent research in social neuroscience (see e.g. [57]). To understand that theory, we first need to

understand the concepts on which it is based. Social categorization is the classification of people into different social groups, which is a deeply rooted human trait, and a person's social identity is the part of the self that is derived from the various memberships we have in social groups. Social identity theory is, therefore, the theory of group membership and intergroup relations based on self-categorization, social comparison and a self-definition regarding in-group properties (i.e. a prototype). Self-categorization is how we categorize ourselves and thereby construct a social identity [58]. According to the minimal group paradigm [59], even explicitly random group assignments trigger discriminatory behavior against an out-group. The idea is that a successful intergroup bias creates or protects (high) in-group status, which provides a positive social identity (which in turn satisfies group-members' need for positive self-esteem). Researchers have successfully explained how groups gain positive self-esteem through intergroup bias but have been less successful when explaining intergroup bias motives due to threats or depressed self-esteem [56]. However, Hogg et al. [58] suggest that competition for positive social identity characterizes intergroup behavior.

When connecting the popularity of having cross-functional teams in the modern workplace (see e.g. [60]) to social identity theory, it becomes clear that it, in fact, decreases intergroup bias. Traditionally, the software engineering industry had more waterfall-type projects where different roles and different phases of project work were, in the extreme case, even conducted by people sitting in separate buildings, which lead to synchronization problems between different organizational functions [61]. Having these various organizational functions share their chores and issues often, would be expected to increase cohesion and understanding of the whole project through shared mental models, which have also gained initial empirical support [55, 62].

Retrospectives – A teamwork practice The aspect of promoting sustainable development with regards to work-hours suggested in principle number eight is partly implemented through the retrospective meetings. The idea is that the team should reflect on possible improvement points about their teamwork at the end of each iteration [63] (as specifically stated in principle twelve). More generally, such reflective meetings are often called *team debriefs*, and have been shown with scientific rigor to increase effectiveness [64]. McHugh et al. [48] found that these types of meeting need work and careful guidance to function in their intended way also in software development. In a recent longitudinal study, Lehtinen et al. [65] showed that, initially, newly formed teams focus more on task progress and task outcome and, as the teams mature, they focus to a larger extent on process and cooperation. Such findings also relate the 'agility' of a team to group socialization and group development since members of the group will behave differently depending on how well integrated they are in the team [54], meaning that a well-integrated individual will be more likely to perform retrospectives in the way they are intended.

Customer acceptance tests – A customer interaction practice As an additional way of involving the customer in the development life-cycle, as the agile principles one, two, and three prescribe, the customer reviews the work after each iteration in a meeting called customer acceptance tests. In

such a meeting, the customer gets to see and test the latest working version of the product with some functionality of high priority to provide continuous feedback, but also to be given the possibility to change the requirements of the product [20]. As mentioned in the introduction, compared to other project management paradigms, the end goal is more likely to change in an agile project since a critical element is a responsiveness to change [7]. The approach prescribes a confession and a realization that, in modern software development projects, finding the perfect goal at the beginning of a project, is a futile endeavor according to Engwall [5] in the management research field, however, having such a mindset could complicate contracting [23]. In the next paragraph, we connect customer interaction to the social identity theory.

Customer access – A customer interaction practice In a practitioner's case study, Krebs [66] reported on the importance of meeting the customer even more often to leverage the desired rapid response to change. Having direct links to the customer have been shown difficult in practices due to shortages of resources provided by the customer [22]. In relation to social identity theory, frequent contact between the technical team-members and the customer would, not only lead to more responsiveness to change, but also a possibility to reduce intergroup bias. The customer representative would, then, be an addition to the team's effectiveness if included on the basis that the roles are defined as different and maintained with their positive distinctiveness when cooperating [67].

Continuous integration and testing – A technical practice The agile principle number nine promotes technical excellence and good design. The practice related to this principle, as suggested by So et al. [40], is the only purely technical practice measured in the survey. It prescribes the continuous integration of source code into the software product that is covered by a set of tests. Such tests are pieces of code written to detect errors in the product code, used to assure code quality [40].

Collocation Opposite to what one might think, the software development processes have gone towards a more analog work-place through agile practices. Having the team collocated in the same room with requirements as sticky notes on physical boards have been promoted by the agile community to, again, increase the velocity of the development in a rapidly changing environment [68]. Many cases have been reported where the communication challenges of distributed teams have been satisfactorily dealt with using modern technology and slightly different practices (see e.g. [69]). Another study showed that both agile and traditional projects have the same issues regarding collocation [70].

Recent social psychology studies have shown that, in general, the team performance on many different tasks is set on group-level, independent of the intelligence of the individuals [71]. The intelligence of groups have instead been shown to be more dependent on social sensitivity (i.e. a person's ability to read emotions in facial expression), and conversational turn-taking (i.e. groups were less collectively intelligent if a few individuals dominated the conversations). In addition, the ability to read facial expression was also a strong predictor

when subjects were communicating through text messages. This result provides evidence that social sensitivity is equally vital in virtual teams, i.e. with emotional sensitivity comes both the ability to 'read the mind in the eyes,' but also the ability to 'read between the lines' [72]. All-in-all, these findings suggest that collocation is not a specific prerequisite for an agile approach to projects.

Summary The agile approach to software development projects evolved in practice and has gained extensive and wide-spread popularity within software development (see e.g. [1]). As have been shown in this review, the prescribed behavior in these agile practices is well-founded in social psychology, which provides social-psychological reasons for their popularity. However, the scientific studies needed to gather empirical evidence from the unique context of agile software development somewhat lag behind.

Hogg et al. [73] explicitly suggest a set of propositions for how social identity and self-categorization relate to the organizational context. One of their propositions is that changes in which out-groups the in-group compares itself to will change the view of the group's own identity, including the properties of the ideal member (i.e. the prototype). In this section, we have described such effects in relation to cross-functional teams and the positive effect of the inclusion of the customer in the development life-cycle, which is in line with what Hogg et al. [73] proposed would happen. Another proposition is that harmonious relations between different subgroups of the organization are best kept by recognizing both the subgroups (e.g. Quality Assurance Engineer, Software Developers, Software Tester, etc.) and other organizational constellations, including the teams and the company as a whole. This proposition means that the cross-functional agile teams must recognize both the value of the team as a whole but also the different roles and make distinctions between them, which is then also true about the customer.

Research gap When looking at the descriptions of the agile practices overall, many of the internal practices seem to assume full group-membership seen from a group socialization perspective [54]. They also assume the entire workgroup to be mature from a developmental perspective [74, 75]. To fully understand the social-psychological components of the team-based workplace in general and the agile context in particular, we also need to investigate the temporal perspective of the interplay between group development and the agile approach to projects. In addition, software engineering provides a context where projects are highly complex and rapidly changing, which is somewhat a new context of research for more general small group research. The existing research on group effectiveness has also been criticized for having been built on a relatively small number of groups from specific contexts [18]. It is therefore essential to apply a temporal perspective to the agile team dynamics, since team agility might be dependent on the maturity of work groups and thus need different and contextualized implementations.

In industry, many agile transitions fail because of lack of team capabilities [76]. Without understanding the psychology of groups, such survey findings are hard to use to improve one's practices. Relating agile practices to deeper psychological theories, like in this thesis, could also provide a deeper

understanding of the psychological processes in the agile workplace. Such recognition would then hopefully lead to better predictability and intervention concerning human factors in agile projects.

1.2.4 Agility and group development research

There are advantages of looking at the group level instead of only individuals and their traits [58, 77]. Such benefits also verified by the few articles found within software engineering and personality that are presented next. In a study by McDonald et al. [78] in the software engineering domain, the authors conclude that as much is derived from the context of the workgroup, as of the people in it. Still, most studies focus on individual psychology, such as studies on psychological needs connected to software development teams (see e.g. [79]). As a side-note, the term 'team' is almost used exclusively for workgroups in software engineering research and practice and denotes a small work-group at an organization. As previously mentioned, human factors have gotten more attention in software engineering [80]. Lenberg et al. [17] also believe more focus on these factors in research is needed. Feldt et al. [81] also argue for the use of personality tests to put together teams, even though they state that personality cannot be considered in isolation. An indication of this is a more recent study by Cruz et al. [82] showing that 40 years of using personality tests in software engineering does not give any consistent results. Furthermore, studies by Licorish et al. [83] and Hannay et al. [83] have shown that personality tests have little predictive value and looking at behavior in context could be a better approach if a prediction is a primary goal.

Melnik et al. [9] showed that people working in agile teams have higher job satisfaction, and one crucial aspect of a mature team is higher job satisfaction [84]. Group work is everywhere, and the flexibility of agile teams makes working in groups even more important [85]. Collaboration in agile teams is described much in the same way as mature groups in group development research [84]. Research also confirms a general gap in taking human factors into consideration within software engineering [11, 17, 86, 87, 88], and we have not found any research connecting the group developmental aspects to team agility.

Another study was conducted by Moe et al. [62], where they studied a Scrum project that was implemented in an organization. They concluded that transition to self-organizing teams needs buy-in from both developers and managers. Team orientation, team leadership, and coordination, as well as the division of work, were factors to consider when moving towards becoming an agile team. The same authors also concluded that challenges when implementing shared decision-making were: alignment of strategic product plans with iteration plans, allocation of development resources, and performing development and maintenance tasks in teams [89]. Hoda et al. [90] also showed the importance of having defined roles in agile teams, which is something stressed as crucial in group development research [84]. The aspects of all these three studies are parts of group development and therefore motivate studying the relationship between the agile practices and group maturity. As also mentioned, there has been some critique of the organizational psychology literature that descriptive group development models are not developed in

context or based on relatively limited observations of a few types of group [18].

1.2.5 The stages of group development

Keyton [42] defines a group as three or more members that interact with each other to perform a number of tasks and achieve a set of common goals. This definition implies that many large groups are in fact a set of smaller subgroups and should be handled as separate groups. If the group consists of more than eight individuals, they are less productive than smaller groups [91]. A 'workgroup' is a composition of members that are striving to create a shared view of goals and trying to develop a structure to achieve these goals. A distinction is sometimes made between a work-group and a team in that a team has found effective ways of achieving its goals [84]. However, in the software engineering domain, a small work-group is often termed a 'team,' and we will use these terms somewhat interchangeably in this thesis even though an agile team is referred to as the end-goal of having built a high performing small work-group in the software development context.

1.2.5.1 Group development over time

The study of the behavior of small groups was launched with the establishment of a research center for group dynamics in 1946, and several research groups proposed different ways of analyzing the behavior of groups [84]. Some studies suggest group development can be described as states or levels of activity, i.e. cyclic models (see, e.g. [92]), but an integrated theory of linear and cyclic models was first introduced in 1964 according to Wheelan [84]. A comprehensive synthesis of various group development models was conducted by Tuckman et al. [74]. The result of their analysis was a conceptual model including of four stages of group development, namely, Forming, Storming, Norming, and Performing. Similar phases of group development were suggested by Agazarian et al. [93] and based on systems-centered theory. The model proposed by Wheelan [84] largely overlaps these stages from those two models and is presented next.

1.2.5.2 The integrated model of group development

Wheelan [94] created an integrated model of group development with four different development stages. These stages are described in more detail in the Related Work section of Paper 2 but are also briefly described here. The key part of the theory is that groups develop through different maturity stages. This fact is straight-forward since we all know we behave differently with people we do not know and people we do know. Furthermore, the developmental stages of groups can also be compared to the developmental stages of an individual human. We figure out what world we got born into (being a child), then we question the structures we see (adolescence). After that, we can organize our lives better and be more informed (young adulthood), and finally, we somewhat find our place in this world and can focus more on how to develop and mature [95].

Stage 1: Dependency and inclusion The first stage is categorized by three main areas; concerns about safety and inclusion, member dependency on the designated leader, and a wish for order and structure. The group is supposed to become organized, capable of efficient work, and achieve goals, so the first stage must have a purpose in getting there [94].

Stage 2: Counter-dependency and fight When the group safely navigated through the previous stage, the group members will have gained a sense of loyalty. As people feel an increased level of safety, they will dare to speak up and express opinions that might not be shared by all members. The second stage of a group's development is, therefore, a conflict phase where a fight is a must to create clear roles to be able to work together constructively. The members have to go through this to be able to trust each other and the leader [94].

Stage 3: Trust and structure The third stage is a structure-developing phase where the roles are based on competence instead of striving for power or safety. Communication will be more open and task-oriented. The third stage of group development is characterized by more mature negotiations about roles, organization, and processes [94].

Stage 4: Work and productivity The fourth and final stage (excluding the termination phase) is when the group wants to get the task done well at the same time as the group cohesion is maintained over an extended period. The group also focuses on decision-making and encourages task-related conflicts. Stage 4 is a time of intense productivity and effectiveness, and it is at this stage the group becomes a team [94].

The most significant contribution of Wheelan is probably to connect a questionnaire to the suggested model of group development (the Group Development Questionnaire [94]). In doing so, it has become possible to diagnose and pinpoint in what group stage the group the group has the most issues, and therefore obtain means to move forward in its development. The whole survey has a total of 60 items and provides a tool for research on, and interventions in, teams.

1.2.6 Project and group life-cycles

It is more and more common to work in the form of projects within organizations, and a project goes through a set of stages that can be described as Idea, Planning, Execution, and Termination [96]. The first stage (the idea stage) is when the idea comes to place, and the company realizes that a project is needed around a specific goal. The planning stage comprises detailed planning, budgeting, scheduling, recruitment, and procurement. The execution stage is when the main project work gets done, and in the termination stage, the work decreases and the results are delivered to the customer.

The group development life-cycle described in Section 1.2.5 and the project's life cycle have a mutual effect on each other, and the problem is that the group's development and the project's development are rarely synchronized. Therefore, the group members could avoid sharing their opinions in the project plan-

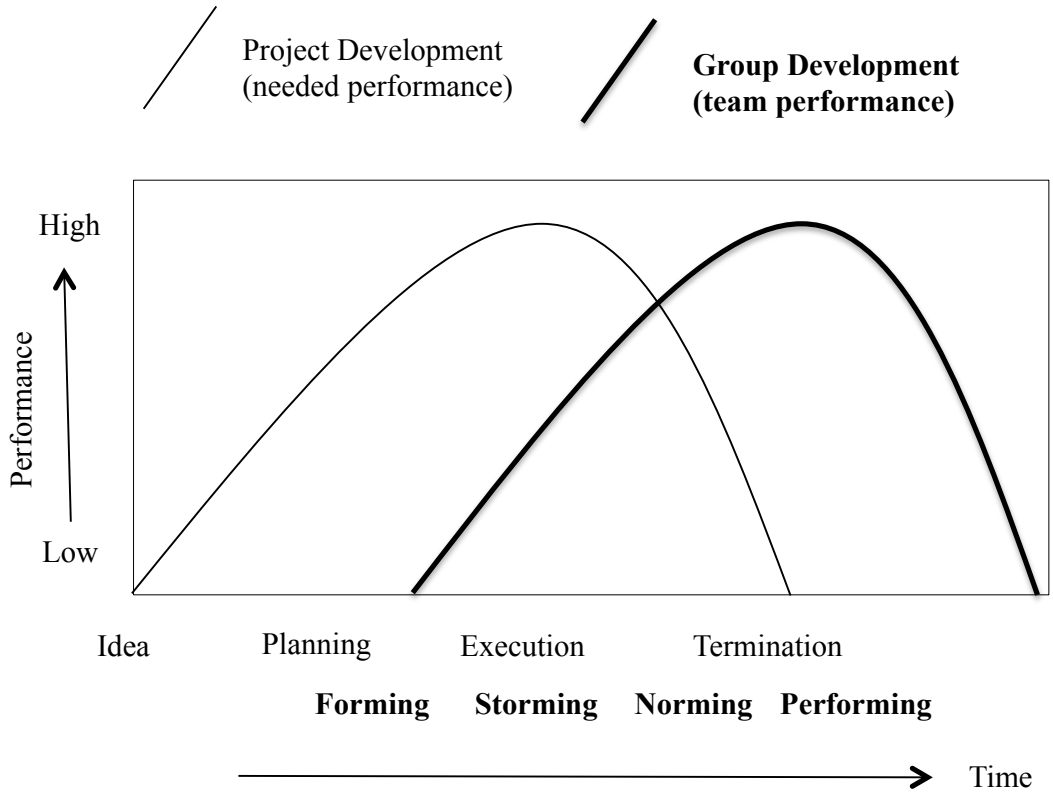

Figure 1.2: Project and group development stages (adopted from [96]).

ning stage, because the group is, psychologically, in the Forming stage. Also, the group might as well be in the conflict (Storming) stage during project execution, which is when the team's performance needs to be at its peak. As the cycle moves along, the team might be starting to perform at its best when the project terminates [96]. All these effects, are, of course, suboptimal (see Figure 1.2).

1.2.7 Situational leadership

Instead of finding an optimal leadership style, Hersey et al. [97] suggested already in the seventies that a leader much adapt and change the style depending on the group. This model consists of maturity levels of group members, but also a balance between relation- and task-oriented behaviors. A leader should act differently depending on the needs of the group. Modern organizational psychology scholars also advocate a dynamic team leadership adapted to emerging needs even in the same situation [98]. The steps suggested by Hersey et al. [97] are illustrated in Figure 1.3.

1.2.8 Similarities in development models

In both the situational leadership model presented in the previous section, the group development model in Section 1.2.5, and the project development model in Section 1.2.6, the phases are divided into a formation stage, a crisis stage, a norming stage, and a work stage. These can also be compared to, for

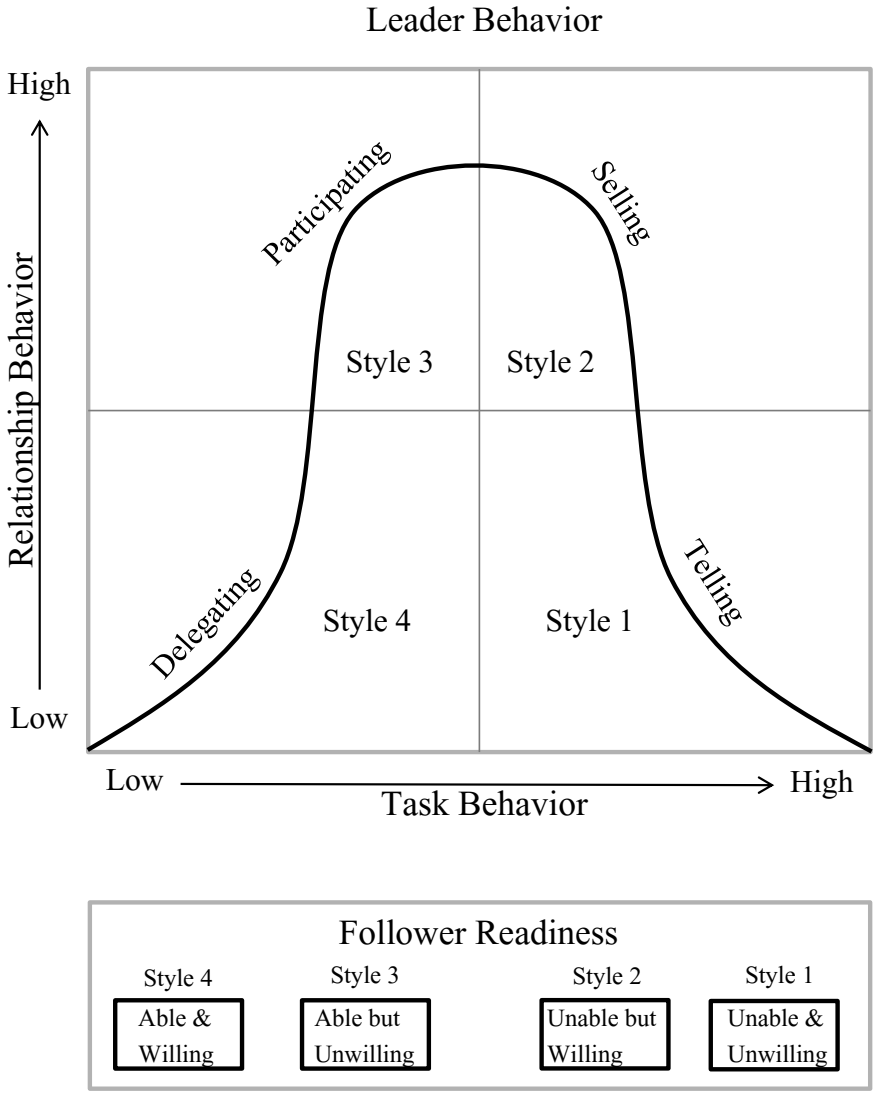

Figure 1.3: Situational leadership (adopted from [97]).

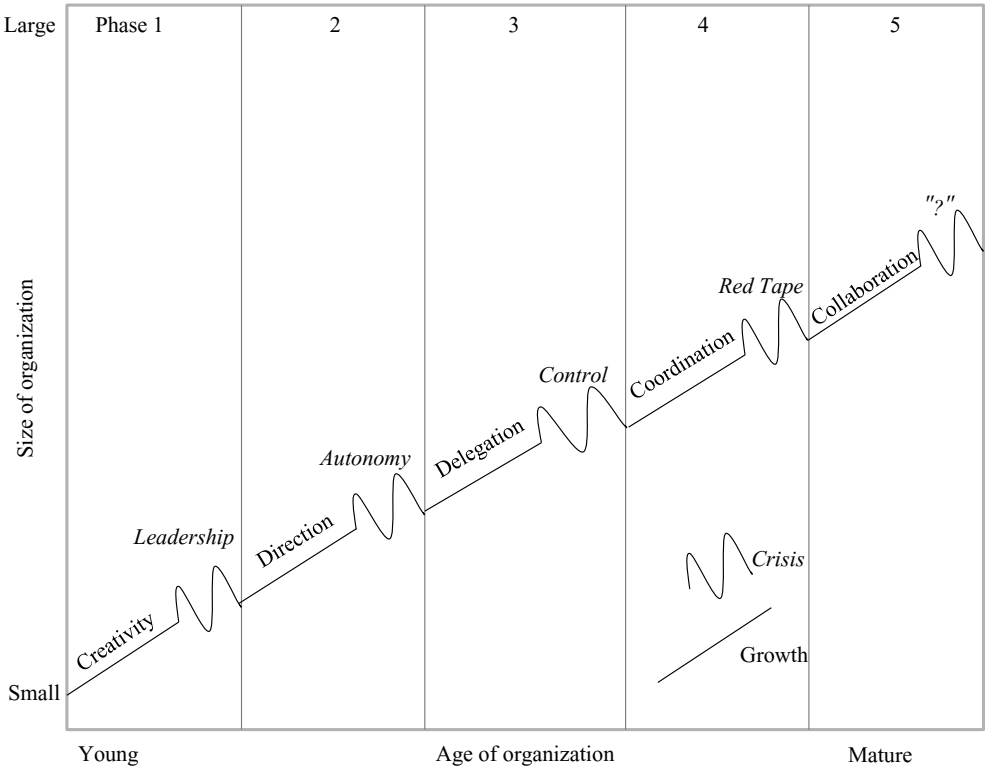

Figure 1.4: Organizational development stages (adapted from [99]).

example, Greiner's [99] model for growing organizations (see Figure 1.4). A new organization starts with the entrepreneurial phase where it grows through creativity. The manager is here individualistic, creative and an entrepreneur. At the end of Phase 1, the organization has a leadership crisis. The following phase is the collective phase where growth is managed through directives that often come from the leader. At the end of this phase, the organization goes through a crisis of autonomy. Phase 3 is a phase of formalization where growth is managed through delegation. Total delegation and autonomy is given from the leader but ends with a crisis of control. The development phase (Phase 4) is to grow by coordination. The leader acts as a watchdog, and this phase often ends up in a crisis of bureaucracy (sometimes called 'red tape'). The final stage is recognized through team-oriented work and interpersonal skills where learning and innovation are present. While we realize that this publication is less scientific, it largely overlaps with other findings in organizational dynamics (see, e.g. [100]). These phases are very similar to the group development stages, human development, situational leadership, and project development. The challenge is to be aware of these and synchronize them carefully.

1.2.9 Agility at three levels of an organization

The final step of this theoretical background is then to connect the three levels of agility to the three more general abstraction levels of organizational theory. As mentioned, there is some evidence showing that agility in its broader sense

is needed at all levels of an organization to reach the intended increase in productivity and it is possible to change the practices on a more superficial level without the cultural change [101]. We, therefore, need the whole organization to be on board with our agile transition, which is not very surprising, but complicated in practice. Returning to Greiner's [99] model of growing organizations (see Figure 1.4), we can see that small organizations, like start-ups, are agile by definition, i.e. they do not have any substantial overhead processes to satisfy when making decisions or negotiating with customers, and are characterized by creativity. However, they will have a leadership crisis sooner or later when the organization gets too large, with the reason being that the founder is then unable to obtain an overview and have control over all operations in the organization. This reasoning provides an explanation to why larger companies can not, and should not, function like small start-ups. However, there are different approaches to growing an organization than the classical command-and-control paradigm, i.e. to instead: 'trust the collective intelligence of the system' [102]. However, such organizations are still rare and out of the scope of this thesis. We also believe it is too early to know the potential of such approaches at a much larger scale since they are more of an exception than a rule today.

Therefore, we need to investigate how agility is connected to maturity on all three levels of the organization, that is, the organizational, team/project, and individual levels. Understanding more about these interactions could increase the predictability of when agile transformation efforts succeed or fail and provide explanations for why. In this thesis, we mostly look at the team-level but also touch upon the explanatory power of investigating individual non-technical skills. Also, we present a study including interviews with the people responsible for the agile transition at an organizational level. The methods used in the papers included in this thesis are described next.

1.3 Method

A well-cited publication in software engineering research is the paper on conducting case studies by Runesson et al. [103]. They also provide an overview of other research methods (case studies, surveys, experiments, and action studies) and their connection to research philosophy. We believe a clear definition of each method and for what research question they should be used, could be counterproductive in research since triangulation with regards to data, methodologies, and theories always increases overall validity. Besides, categorizing data collection techniques as qualitative or quantitative could be misleading since a 'qualitative' data collection technique (like, for example, interviewing) might sometimes be used to collect quantitative data (counting specific words, for example).

Therefore, this section will first give a more philosophical reflection on scientific discovery and then present what methods were used for each appended paper more in relation to types of data.

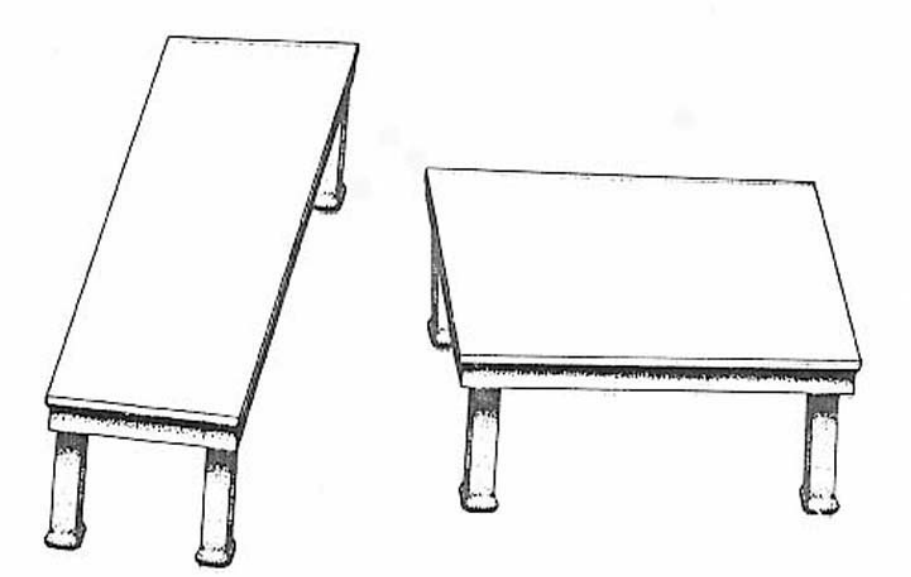

Figure 1.5: The two tables illusion. Are they of different size?

1.3.1 Research – The search for truth

Human perception One aspect that should be considered, when investigating human-beings in any environment, or 'system,' is that we can never investigate the real world using people's reasoning. We can not investigate the real world in deterministic mathematical models either. However, these models are often closer to the real world. If we use people as research subjects, we will most often only research their perception of the context and rarely the context itself. This difference is an important distinction to make because people can have alternative interpretations of the same situation (something witness psychology must deal with [104]). Sometimes we all make the same assumption about the world, and we all can get tricked in the same way, like the tables in Figure 1.5 (which are of the same exact size). Besides, we sometimes automatically assume different contexts, like the dress shown in Figure 1.6. The people that see the dress white and gold assume that dress is outside in natural lighting and the ones seeing the dress in black and blue automatically assume the dress to be indoors in artificial lighting.

The point we want to make is that different people have different perceptions of real objects. Even time (which is often perceived as an exact metric) is different depending on where the observer stands (as theorized by Albert Einstein and first proved empirically by Hafele et al. [105] in 1972). When investigating human factors in software engineering, we should always state that we describe the perception of the construct and not the construct itself. The perception often what we are interested in any way.

Memory and motivated reasoning To provide some more detail of how memory works, a brief overview of research on the motivated reasoning in social psychology is provided next.

Our old view of humans as 'rational' concerning objective information has largely been abandoned in the modern psychological discourse [43] and

Figure 1.6: The dress picture. White and gold, or black and blue?

we know that 'feeling' and 'doing' are inseparable processes in the human brain [106]. Kunda [107] suggests that motivation (or goals) is the source of why humans reason in a biased way and attributes this bias to our reliance on a set of affected cognitive processes. These processes are strategies for accessing, constructing, and evaluating beliefs. She makes a distinction between motives to arrive at an accurate conclusion and motives to arrive at a particular (or directional) conclusion. Kunda argues that: "both kinds of goals affect reasoning by influencing the choice of beliefs and strategies applied to a given problem. But accuracy goals lead to the use of those beliefs and strategies that are considered most appropriate, whereas directional goals lead to the use of those that are considered most likely to yield the desired conclusion" [107].

People are generally motivated to be accurate and are aware of the effort-accuracy trade-off, meaning that people do not only assess how good an outcome they desire but also the amount of cognitive effort that is needed for such an outcome. In an experiment by McAllister et al. [108], the authors found that subjects that were motivated to be more accurate chose more complex and time-consuming decision-making strategies than those who were less motivated. These manipulations were done to show that increased accuracy motives lead to a reduction of bias and that many biases are due to hasty reasoning. However, Kunda suggests that for accuracy to reduce bias: "it is crucial that subjects possess more appropriate reasoning strategies" [107], implying that bias can still occur even if motivation exists, and argues further that such treatments do not explain the impact accuracy goals have when accompanied by directional goals. People attempt to be rational, and when

subjected to directional goal, they construct a justification of the desired outcome, according to Kunda [107].

Kunda [107] states that people draw the desired conclusions if they can generate evidence to support such a conclusion through a biased memory search driven by their goals. Just like Pyszczynski et al. [109], she argues that there is a self-esteem motive to being able to defend the conclusions by conducting a biased hypothesis testing equal to the concept of a biased memory search. In a study by Kunda et al. [110], subjects selectively responded to a personality test depending on what traits had been promoted for a successful career, presumably because they wanted to view themselves as characterized by higher levels of such desired traits. This result means that we even tend to change our concepts of ourselves when motivated to do so by such a directional goal. However, such effects were mediated by prior self-knowledge, meaning that we can only bias our memory to a certain extent in relation to justifying such conclusions. In an additional study, subjects reported less frequent behavior in relation to tooth brushing depending on if it was primed as good or bad for health [111]. Kunda [107] concludes that the accuracy goals lead to more intense processing, but at the same time, the directional goal creates bias. We will come back to motivated reasoning in the validity threats section of this thesis.

The research conducted in this thesis The main point we want to make in this section is that we need to understand people's motives when investigating human factors in software engineering. It is sometimes frustrating to have realized that our research is only a fraction of the research needed to generalize to the intended population of 'agile teams' across the globe, however, adding group maturity to the concept is still, though only an initial, but an utterly important step, we believe. We are far from having proven any concepts and the big scientific discoveries of the truth for our world, like the theory of evolution [112] and heuristics in behavioral economics [113] (in some contexts), we consider true at this point, simply because researchers have gathered massive empirical evidence.

We would like to take the opportunity here to refer to the quote by Stephen Pinker at the beginning of this thesis. It has been our purpose to consistently try to debias our research by being aware of existing cognitive biases. One such attempt was to include a paper investigating the individual micro level of nontechnical skills to suffer less from confirmation bias when suggesting the addition of group maturity to the understanding of agile teams.

1.3.2 Methods used in this thesis

The statistical method of factor analysis presented in Paper 1 (Chapter 2) is one way of checking if the data support the idea that a test measures what we hope it does. That method will not replace other aspects of validity, but we do not see any disadvantages with collecting empirical evidence for surveys used in research as one step in validating measurement scales used. In the field of psychology, researchers need to be very careful stating that surveys that have not been scientifically validated give any evidence for a certain research hypothesis. We believe the field of software engineering should be as careful

when using poorly validated tools both in research and practice. Paper 1 (Chapter 3) uses such a statistical test (i.e. a factor analysis) on an agile maturity model, showing that it is not enough to develop an agile measurement by only using anecdotal evidence or data from only one case.

Of course, collecting a lot of data can be cumbersome. Sometimes we need to do as well as we can, given small samples and scarce information. Also, if a research angle is new and unexplored, it is impossible to know what questions to ask in a survey. For these new emerging fields or aspects, a qualitative approach is the only option before one can triangulate the construct with additional quantitative data, and finally, conduct experiments. That is why we conducted, and added qualitative data (in addition to more quantitative data), to the research presented in Paper 2 (Chapter 3). We believe that triangulation is key to increasing the validity of smaller studies.

Paper 3 (Chapter 4), though, did not apply triangulation. Instead, we used already quite rigorously validated scales created by other scientists. In doing so, we relied on their previous findings and only contributed with a connection between data from their scales. To changed perspective, Paper 4 (Chapter 5) connects group development data to that of external measurements of productivity, namely software development velocity and planning effectiveness. However, with such a small sample we also added a set of interviews to increase the validity of that case study. As mentioned earlier, we wanted to change the level of analysis in Paper 5 (Chapter 6) in which we collected a larger data sample to increase the statistical power, i.e. the probability of finding an effect, if there is one.

1.4 Chapter/Paper summaries

In this section, we will summarize the different papers and state their contribution. After that, a discussion of these results will be presented.

1.4.1 Chapter 2/Paper 1: The prospects of a quantitative measurement of agility: A validation study on an agile maturity model

To investigate how agile practices and their adoption are related to agile teams, we need to measure agility somehow. The issue of how to measure agility with high validity is not thoroughly researched in software engineering (see Related Work in Paper 1). Therefore, as a first step of assessing agile maturity measurement, Paper 1 presents a validation study on such a measurement tool (including a pretest case study with a team). The method used for evaluating the tool was taken from how scales/measurements are developed in psychology with a focus on rigorous and well-used statistical tests. The results show that the tool under validation needs more work and further validation. However, we also discuss the difficulty of measuring agility in such a way.

The main contributions of Paper 1, as stated in the highlights of the publication, are:

1. A quantitative measurement of 'agility' with connected confidence intervals to the items (developed in the pretest).

2. A positive result from practitioners on that quantitative agility measurement tool.

3. Validation tests of internal consistency and construct validity (negative results).

4. New groups of item are presented, but we generally question the usefulness of such agile maturity models.

5. A described tradeoff between quick quantitative versus time-consuming contextual assessments.

This paper helps to answer RQ1: "Is the agile adoption framework is valid according to quantitative tests for internal consistency and construct validity?" and the answer was negative. Some published papers in software engineering make use of these statistical validation techniques in their methods (see e.g. [10, 40]). However, we conclude that these methods are useful for larger survey research in software engineering generally, and specifically to find a measurement of agility. In the quest to find measurements of agility, studies with larger samples sizes and applied statistical validation techniques, like that of a factor analysis, are a must to move forward. However, we also conclude that the concept of 'agility' needs to be broken drown to find valid scales for whatever sub-parts of 'agility' that are of interest to the researcher or practitioner. Therefore, in the next paper, we only looked at team agility from a group-psychological perspective.

1.4.2 Chapter 3/Paper 2: Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies

Paper 2 directly aims at describing the connections between group maturity and team agility. This perspective of agility helps define the vague concept of agility somewhat since a mature team in social psychology and agile teams go hand-in-hand. Therefore, we see strong indications that it would help both researchers and practitioners to view agile teams as mature teams since it provides a path to maturing, and through that maturation, becoming more agile. Then many of the agile practices can be seen as enablers of group development, which leads to a higher understanding of what happens in teams as well as increases the predictability of behavior.

The main contributions of Paper 2, as stated in the highlights of the publication, are:

1. A found correlations between measurements of agility and group maturity.

2. An in-depth interview analysis of how these correlated factors function.

3. It helps defining agility as a mature group (as described in group psychology).

4. It suggests how agile teams can work with group development.

Paper 2 comprises ten semi-structured interviews, and the reason for conducting those was to investigate the causal relationship and conduct a more in-depth analysis of how group development and agility are connected. Such an analysis provided a greater understanding of what these managers and agile coaches do practically in their daily work regarding the human factors of building agile teams. This result implies that teams adopted agile practices differently depending on their group development stage, which makes it possible to suggest strategies and support for agile implementations in connection to these stages. Such guidelines do not exist today, to the best of our knowledge.

Paper 2 helps us to answer RQ2: "How is group maturity connected to building agile teams?" and the answer was affirmative; they are intimately connected. We are the first researchers to provide empirical evidence of connections between agile teams and group development. The study shows how software engineering could use knowledge from social psychology instead of reinventing the wheel. Groups have been researched in psychology for almost a century and to focus on where software engineering is different instead is a better investment, in our opinion. We also stated that many practitioners evidently work on group development to help teams mature in their agility; however, providing empirical support for their work and create a scientific case of their work's importance, is the main contribution of Paper 2. The next paper investigates the connections between agile practices, interpersonal conflict (apparent in stage 2 of group development), and perceived productivity.

1.4.3 Chapter 4/Paper 3: The links between agile practices, interpersonal conflict, and perceived productivity

Paper 3 is an investigation of the hypothesis that agile practices should be negatively correlated with the second stage of group development, namely the conflict stage called 'Counter-dependency and fight.' The paper helps to answer RQs 3a: "Which, if any, agile practices are positively or negatively associated with interpersonal conflict?" and 3b: "Which, if any, agile practices are positively or negatively associated with perceived productivity?" The first concerning the conflict stage and the second being about the general validity of measuring agile practices as suggested by So et al. [40]. The results showed that interpersonal conflict was negatively connected to the agile practices 'Iterative development' and 'Customer access.' Regarding RQ3b, the agile practices 'Iteration planning' and 'Iterative development' were both positively correlated, and the practice of 'Continuous integration & testing' was negatively correlated, to the perceived team productivity by agile team members.

The main contributions of Paper 3 are:

1. Evidence of the connection between the conflict stage of group development and some of the agile practices.

2. Positive links between some agile practices and perceived team productivity.

3. Negative connection between 'Continuous integration & testing' and perceived team productivity.

4. The highlighting of the importance of teaching agile teams good conflict resolution techniques.

In summary, the paper contributes mainly with an affirmative result to the hypothesis of existing connections between the intended use of agile practices and developmental patterns of groups concerning interpersonal conflict. We, therefore, have collected additional evidence of the importance of adopting and understanding agility of software engineering teams from a group developmental perspective. We also found that some agile practices contribute to team productivity and some do not. Paper 3 did not include any external data on productivity, which is why we conducted the study that is presented next (Paper 4).

1.4.4 Chapter 5/Paper 4: The connections between group maturity, software development velocity and planning effectiveness

Paper 4 presents a correlative study on all the group development stages, development velocity, and planning effectiveness. It, therefore, provides us with answers to the RQs 4a: "What is the association between group maturity and planning effectiveness?" and 4b: "What is the association between group maturity and software development velocity?" The former was a found positive association and the latter was that no association could be found.

Nineteen developers from four different teams participated and 16 of those also took part in semi-structured interviews. In order to compare the velocities of different teams, we used scrum tasks instead of points since the company assessed them as being of equal complexity across teams. Therefore, the velocity was measured as the number of hours a team spent on scrum tasks on average per sprint, which should be interpreted as an inverse mean velocity.

In the participating teams, scrum points, on the other hand, were given to teams when the product owner approved finished user stories. To obtain a measurement of effectiveness (delivering value) as opposed to efficiency (working fast), we calculated the ratio of planned points at the beginning of a sprint over earned points at the end of each sprint. Planning could be seen as a more traditional approach to projects, but in fact, sprint planning is of utter importance to deliver value in agile projects. The difference is the length of the planning horizon, i.e. in an agile project, planning is conducted in much shorter iterations.

The results showed that the Stage 4 'Work and productivity' scale of the GDQ was strongly correlated to planning effectiveness but not to velocity, meaning that delivering what is needed is dependent on the group development, but not the ability to work fast. These quantitative results were confirmed in the qualitative data analysis.

The main contributions of Paper 4 are:

1. A first in-depth qualitative and quantitative study on group development and software development performance.

2. It provides support for the connection between group maturity and the planning effectiveness of agile teams.

3. It highlights the fact that efficiency in software development teams might be more dependent on individual technical skills than team dynamics.

4. It suggests that teams in the software engineering context need to be given the possibility to mature over time from a group development perspective to achieve higher planning effectiveness; thus, becoming a self-organizing unit where all members can provide input for accurate sprint planning.

In summary, we found support for the group development model in relation to external measurements of productivity as well. However, the group maturity was not dependent on the software development velocity, which might indicate that working fast is very dependent on individual technical skills in the software engineering domain. To also investigate the micro level of analysis, we conducted the study presented next (Paper 5).

1.4.5 Chapter 6/Paper 5: Non-technical individual skills are weakly connected to the maturity of agile practices

To validate our assumption of explaining team agility on a group-level, Paper 5 investigated if team agility could instead be explained by the individual nontechnical skills of the team members. Paper 5, therefore, helps us answering RQ5: "Are individual nontechnical skills connected to the mature use of agile practices?" and the answer was that they are not. Through collecting data from agile team members, we created a set of multiple linear regression models using individual nontechnical skills as factors and the agile practices, one-by-one, as response variables. The results showed no or very low associations between the measurements, which made us conclude that individual nontechnical skills are not the optimal level of analysis when trying to explain team agility. This result, therefore, supports our previous papers' assumption that agile teams need to be understood from a group dynamics perspective, and not an individual one, i.e. agility is a team capacity and not the sum of the individual team members' nontechnical skills.

The main contributions of Paper 5 are:

1. A larger quantitative study ($N = 197$) on the relationship between individual nontechnical skills and agile practices in use.

2. It shows lack of support for the assumption that team agility can be understood by looking at team members' individual nontechnical skills.

3. It provides evidence of the importance of focusing on the nontechnical skills as a team-level capacity instead of assuring that all individuals possess such skills.

4. It shows that, just like the collective intelligence has been shown to be unrelated to individual intelligence, individual nontechnical skills seem to be unrelated to nontechnical team skills.

In the next section, we will discuss the presented papers in relation to the background and related work, and see how they together answer the overall

research question of "How is group maturity related to team agility in software development?".

1.5 Discussion

The results of this thesis show that there are connections between a group's maturity and what is meant by an 'agile team,' i.e. this thesis contributes with showing the relevance of psychological group processes when building agile teams, which has not been researched prior to our work, as far as we know.

The agile approach to software development, but also to projects in general, is based on acknowledging that the human and social factors in projects are among the critical success factors [21]. When noticing their importance, some software developers created the 'agile manifesto,' in which they tried to specify what principles teams of software developers should follow to succeed [4]. These principles came from practice and were therefore not created in reference to existing management, organizational or social psychology research (see e.g. [114]). As can be concluded from the literature review of this thesis, many of the agile principles are far from new in relation to human knowledge of work groups, like the 'retrospective' [65] that is called 'team debrief' [64]. However, what might be considered as having a stronger acceptance in agile as compared to other paradigms, and maybe due to its inclusion in the agile manifesto, is the implementation of responsiveness to change [4]. The software industry is a context of rapid change in complex projects, which enforces the necessity of dealing with volatile demand [7]. The reasons for not relating agile software development to any existing science might be due to lack of research knowledge from practitioners, but also a strategy for selling expensive consultancy hours (i.e. selling an entirely 'new' management paradigm is simpler than selling small changes to an old one). This explanation, however, is pure speculation from our side.

As we have also seen in the literature review of this thesis, there is a lot of overlap between existing knowledge of, and research on, the work-place in general and the agile practices. A few internal organizational examples being decreasing inter-group bias through cross-functional teams [60], striving towards self-organization of teams in order to increase responsiveness to change [46], creating organizational citizenship behavior through shared visions [50], empowerment and trust [47], and removing 'waste' in the process [53]. Externally, having good relationships with the customer through participation and access has also been shown to be valuable from a social-psychological perspective [73]. The only practice that we have not found any specific support for in the literature review for this thesis was a connection between an agile approach specifically and the practice of 'collocation.' While we found that being collocated always makes the social relationships easier, we did not find this to be any different for agile teams as compared to other types of team.

To put the 'agile team' in context, studies have shown the importance of the whole organization adopting the agile principles, not only the teams (see e.g. [101]). What is interesting with the group development questionnaire is that the organizational environment can hinder the group's development, i.e.

a group might not get a chance to mature, but the causes could be external. This effect poses an essential realization in practice since poor results on the GDQ is tempting to use as a team evaluation tool. However, it is apparent in the consultancy guidelines (not publicly available) that the workgroups own their result and to use it for management objectives is strictly prohibited. Based on our studies on group development and team agility, we have seen that the micro-level is much less useful than previously thought, which is also confirmed by negative results of using personality tests in research on software engineering teams (see, e.g. [82, 83]). The strategic (or macro) level has been shown to be important in previous research [13], and the research included in this thesis has shown that agile maturity needs to be investigated as group phenomena. The building of teams implies that 'team agility' needs to be understood both in context, i.e. the organizational ecosystem in relation to organizational maturity [100], but also in relation to project phases [96], and individual team members' readiness in relation to types of leadership [97]. This thesis then adds the temporal perspective of the maturity of the work groups, i.e. the answer to the overall research question of "How is group maturity related to team agility in software development?" is that:

> Workgroups at different group development stages will adopt team agility differently even if they are all given apparently the same opportunity within the organization.

Our results imply that a more thought-through approach to agility, that also includes aspects of group maturity, could increase the understanding and predictability of the agile implementation. The results suggest that the behavior observed in the agile practices are less frequent (or used differently than intended) in immature groups. This result implies that leaders and managers could be helped by preparing for this disparity and accept this as a natural part of a group's development. Practitioners should be aware of that the group could make different use of the agile practices when less mature and first gets the possibility to leverage these practices when the group has developed further. Even if some practices will be used differently in the first compared to the latter stages of group development, this does not imply that these practices should not be implemented in the beginning. Rather, such practices are most likely needed for the group to develop since, e.g. Retrospectives, give the group a forum to reflect on group work. Such reflection is needed in all stages. However, the content will differ since the group has different issues to work on from a group developmental perspective. This temporal perspective is also in line with the results by Lehtinen et al. [65] where they observed behavioral changes in retrospectives over time.

For example, a group with lower group maturity might be less likely to use some of the practices in the way they are described. The opposite is then also true, meaning that a group with a higher group maturity score might behave more in a way described in the agile software development literature. Such results contribute to our understanding of why a transition to a more agile approach is more difficult for some teams at specific points in time. The agile literature mostly includes success stories of how agile teams work when

they work exceptionally well (see e.g. [33]). One could argue that most suggested processes of effective teamwork assume mature teams, but it is evident that there is more of a team focus in the agile approach compared to other approaches to projects [85], i.e. group dynamics is more important to understand in the agile team than traditional ones.

The fact that groups have different needs over time could be connected to leadership research that focuses on that different leadership is needed in different contexts depending on what the group and group-members need. The situational leadership model [97] includes maturity levels of the group members, but also that balance is necessary between relation- and task-oriented leadership behavior. They present the four different styles 'telling', 'selling,' 'participating,' and 'delegating' that are somewhat translatable to the group development stages. With more immature groups, telling and selling are needed approaches for the leadership to be successful, while at the more mature stages, participating and, finally, delegating are more effective styles, since the group can self-organize. This means that the agile practices need to be implemented using a different leadership style depending on the maturity level of the group. In the agile method Scrum, we find descriptions of 'agile leadership' as being facilitating instead of directing [29,32]. Facilitation works well in a mature group, but if that is not the case, the leader will need to behave differently to move the group forward. That is why situational leadership adapted to the group development stages needs to be incorporated into software engineering processes, and if they are not, leaders and managers will wrongfully try to follow a method that could be hindering the progress of that specific team.

It is also cumbersome to only look at the process maturity, like CMMI (Capability Maturity Model Integration) or the ISO/IEC 15504 SPICE (Software Process Improvement and Capability Determination), since we want agile value-driven organizations that use agile practices to implement agile principles. Also, process maturity models are based on building customer trust by process infrastructure instead of working software, and customer participation [115]. The strategists (managers/leaders) and the employees of an organization need to set the vision according to the organization's purpose of existence in alignment with the agile principles (the cultural change) and then select agile practices to support that journey [4].

1.6 Validity threats

Most reviewers and authors in the software engineering research field, base their concept of validity on two publications, namely [116] and [103]. While we highly appreciate the authors' work in these publications, we are afraid the four categories of validity threats promoted for both experimentation and case study research (i.e. construct, internal, external, and reliability), have often turned into a 'quick fix' at the end of papers. This tendency might be because the categories are not described in more detail concerning test theory. In this section, we suggest a different approach to validity threats than what is commonplace in software engineering research and apply that approach to the appended papers.

A stepwise and shallow presentation of these four validity threats categories, at the end of each paper, indicates an immature approach to validity aspects in the software engineering field, even for well-designed studies regarding the resources available and novelty of the work presented. Besides, we have met very few researchers who remember the name of the first category and what it comprises, i.e. construct validity, simply because researchers have read other categories in related fields, such as criterion, conclusion, concurrent, content, and convergent validity. Sadly, we also lose many important subcategories of construct validity that should be taken into account throughout any research project or program, especially the ones promoting validation with real data. Such studies are of course demanding and require a lot of resources, which imply that they take time.

In accordance with Feldt et al. [117], we believe that "validity is a goal, not something that can be proven or assured with the use of specific procedures." There are also trade-offs between different validity threats, e.g. if more resources are spent in assuring a good operationalization of a particular cause and effect, the whole experiment might suffer from being a 'toy problem' with very little or no practical value. Therefore, we do not agree that as many validity threats as possible should be listed at the end of each publication together with statements of how they were mitigated, since such a utopian study does not, and will never, exist. Instead, we suggest that the *method* section should be reworked until it is as clear as can be. With a well-written method section, threats to validity will be clear in the description of the planning conducted, and statements regarding sample sizes and generalization need not be explicitly stated or repeated in a discussion of validity threats at the end of each paper. As Feldt et al. [117] also conclude, the guidelines given in both [116] and [103] are from conducting experimental planning or case studies. They were not intended as post-study 'quick fix' checklists by the authors, which they also explicitly state in [103]: "It is, as described above, important to consider the validity of the case study from the beginning."

We suggest that no categories be used for listing validity threats in research papers since we believe they are somewhat counterproductive, despite the fact that we have most often done that ourselves. The more complex validity aspects that need clarification after the reader has read the method section could be discussed under a section called 'Validity threats,' 'Threats to validity,' 'Limitations,' or the like. Another option is to simply write the threats as a part of the discussion section, but the practical significance of the threats needs to be in focus. What is important is that both authors and reviewers consider threats to validity throughout their research. To help to implement such an awareness, we suggest a checklist below inspired by seminal work conducted by Messick [118] in relation to testing validity, but where we also include validity aspects in relation to specific research studies as already suggested by [116] and [103].

Messick [119] defines validity as follows:

> "Validity is not a property of the test or assessment as such, but rather of the meaning of the test scores. These scores are a function not only of the items or stimulus conditions but also of the persons responding as well as the context of the assessment. In particular, what needs to be valid is the meaning or interpretation of the score;

as well as any implications for action that this meaning entails."

This definition implies that we always validate the usage of a test, and never the test itself. We would like to mention here that a 'test' refers to a psychological test, i.e. a measurement of a construct (a 'construct' in any scientific field is a phenomenon defined as a distinct category that is under study). In the quote by Messick, we can see that he advocates a more applied and practical treatment of validity. He also argues that validity of a test is only one construct that he calls 'construct validity.' He writes that different aspects of construct validity can still be presented in order of convenience. However, they are still interrelated both operationally and logically. Also "the principles of validity apply to all assessments, whether based on tests, questionnaires, behavioral observations, work samples, or whatever" [119]. The presented six aspects are Consequential, Content, Substantive, Structural, External, and Generalizability, and, to clarify, are all concerning the actual measurements, and not, for example, the generalizability of treatment in a specific experiment.

Besides, reliability is seen as a prerequisite for validity, and the external, internal and conclusion validity in relation to a research study is also included below, following [116] and [103]. However, we think it is of utter importance to change the culture in software engineering research from seeing tool-constructing as the holy grail of research and instead value validation studies higher. To build theory, and make good use of research funding, software engineering researchers need to conduct; ideally, a study of each aspect of construct validity presented below, before drawing conclusions to the intended population and connections between constructs. Therefore, throughout a research project or program, we suggest the following checklist be used:

- Reliability — Reliability is 'repeatability' or 'consistency'. A measure is considered reliable if it would give us the same result over and over again (assuming that what we are measuring is static), essentially answering the question: Does the test measure anything?
    1. Stability – Is the testing stable if we do a test and then another test on the same subjects under the same conditions (test-retest procedure) or parallel testing?
    2. Internal consistency – Is the test consistent with regards to, e.g. the Kuder-Richardson Formula 20 (KR20) [120] or Cronbach's α [121]?
- Construct Validity — Does the test measure what it is meant to measure?
    1. Consequential – What are the potential risks if the scores are, in actuality, invalid or inappropriately interpreted?
    2. Content – Do test items appear to be measuring the construct of interest?
    3. Substantive – Is the theoretical foundation underlying the construct of interest sound?
    4. Structural – Do the interrelationships of dimensions measured by the test correlate with the construct of interest and test scores? These can be tested by using a factor analysis (FA) or a principal component analysis (PCA).

5. External – Does the test have ecological, convergent, discriminant, and predictive qualities?
   (a) Ecological — Is the real-world behavior in accordance with how a subject answers the test?
   (b) Convergent — Is the test similar to (converges on) other operationalizations that it theoretically should be similar to? For example, two agile maturity models should result in similar levels of maturity when applied to the same organization at the same time.
   (c) Discriminant — Is the test dissimilar to (diverges from) other operationalizations that it theoretically should not be similar to? Maybe we hypothesize that testing practices should not be correlated to developers' self-esteem. If so, measurements of both should have a low correlation.
   (d) Predictive — Can the tests predict something it should theoretically be able to predict?
6. Generalizability – Does the test generalize across different groups, settings, and tasks? This aspect of generalizability is in relation to the actual measurement only, i.e. generalizability of one measured construct only. This category is not to be confused with external validity of the whole study described below.

- Conclusion validity — Is the degree to which conclusions reached regarding relationships in our data reasonable?

- Internal validity — Internal validity is most often seen as an investigation of causality between measured constructs, i.e. could there be other confounding factors in the study where the test(s) is(are) used?

- External validity — External validity is concerned with generalizations in relation to the *whole* study. Are the results valid for the intended larger population? In the case of experimentation, this includes aspects like treatment, subjects, and context. In the case of a correlative survey study, the external validity would be the generalizability of the actual correlation study and not the constructs being measured. In the ideal case, a smaller correlation study deploys measurements that have already been validated with large datasets.

We believe such a systematic approach to threats to validity is useful in research. However, some troublesome studies about research evaluations show that researchers, just like all people, often conduct a biased memory search based on their motivation. However, people are not biased without the feeling of being able to justify their conclusions. In a study by Gilovich [122] the loss of a football team was explained by a fluke in the game only if the subjects were aware of it, and if they were not, they lost faith in their team's talent. There is also evidence showing that directional goals may affect the use of statistical heuristics. In a study by Ginossar et al. [123], they showed that subjects used information (like a base-rate) only when it could motivate reaching their directional goal. By re-analyzing such results, Kunda [107] proposes that the

infamous results by the Nobel Prize winners of economic sciences, Tversky and Kahneman popularized by Kahneman [124], is an oversimplification and that the more analytical reasoning is also prone to bias. Kunda [107] also shows that the more generally known confirmation bias [125], is not purely a cognitive bias, but better understood as a biased memory search (even if Kunda herself did not use the term 'confirmation bias'). In a troublesome review of biased research evaluations, she presents cases where subjects in favor of a specific directional goal judged research studies as of higher validity and better conducted as compared to subjects with a different predisposition (see, e.g. Pyszczynski et al. [109]). She concludes that the subjects used heuristics depending on the conclusions of the research, not its methods [107], which is, again, utterly troublesome for the academic community. The question is if we will ever be able to manage confirmation bias in research. At least, we can be structured and consistent in our requirements of the different scientific methods used. Hence, instead of listing categories of validity threats, we will below describe the actual threat we see to the different papers in this thesis.

The confirmation bias problem in research has also been shown to exist in software engineering explicitly [126], which is far from surprising, but important to show. On top of confirmation bias comes both researcher bias (statistically non-significant results that become significant through questionable research or analysis practices) and publication bias (statistically non-significant results that are not reported), something Jørgensen et al. [127] also have studied in the empirical software engineering domain. All these studies show that we need a more systematic approach to validity on many different levels.

Generalizations need to be drawn with more care, we argue, and even the most extensive empirical studies in software engineering have a too small sample size to state anything about the 'truth' for these concepts. It takes a field decades to build up a body of knowledge extensive enough for a meta-study to have such claims of external validity. See for example Freeman et al. [128], where they used 225 studies to conclude that active learning outperforms traditional lecturing with regards to student performance. Or the conclusion that we inherit 49% of our developed human traits, based on 2,748 publications including 14,558,903 twin pairs [129]. The point is that new concepts in exploratory research are most likely not possible to generalize outside of the specific case. We could choose to believe that it is true somewhere else, but without empirical evidence to support such a claim. However, the internal validity is often considered higher in interview studies since a validation of the possible causal relationships is included in the design.

1.6.1 The validity threats of the appended papers revisited

Below, we reanalyze some of our publications using our proposed checklist. The outcome is the same as the validity threats sections of each paper except Paper 1 (Chapter 2) which is a validation study itself. Our new checklist provides us with an overview of where in the overall validation process we currently are, which can guide further validation studies in our research. We can also use the checklist to discuss the threats to validity when we opt to use the specific test in, for example, an experiment.

The first paper we will reanalyze, Paper 1 (Chapter 2), shows one aspect of testing the validity of an agile maturity tool. The standard validation method used can easily be repeated on another data set and is a well-used and validated procedure. It is easy to confuse the validation of the measurement and the validation of our validation study. We only analyze the former below since we see our study as a step in a more extensive validation process of that specific measurement tool. In our reanalyses below we opted to use a 5-point scale ranging from very low to very high. The score of an entire category is not solely based on an average, but instead a qualitative assessment of the impact of the conducted validation but also in relation to if a validation aspect has not been investigated at all. The exact values are not essential, what is important is the guidance the checklist provides. Also, just like Wholin et al. [116] concludes, not all categories of threats will be useful for all studies.

- Reliability — **very low**
    1. Stability — **not tested**
    2. Internal consistency — **very low** due to our study.
- Construct Validity — **moderate**
    1. Consequential — **moderate**, since we see some contradictory results between the statistical validation results and the qualitative data collected by both Sidky [130] and us. The possible misinterpretation of scores, therefore, need more investigation.
    2. Content — **very high**, based on Sidky's [130] work.
    3. Substantive — **very low**, because agility is defined as the use of practices and that the tool mixes many different abstraction levels of the organization.
    4. Structural — **very low** due to our study.
    5. External — **low** (too many not tested).
        (a) Ecological — **high**, since we also conducted a pretest with a focus group in relation to their own work context. Sidky [130] himself also included validation with practitioners in his original work.
        (b) Convergent — **not tested**
        (c) Discriminant — **not tested**
        (d) Predictive — **not tested**
    6. Generalizability — **moderate**, since it was assessed by agile practitioners by Sidky [130], but only in one study at one point in time.
- Conclusion validity — **low** since a threat to our validation study is the relatively small sample used for the validation procedure and the fact that we changed the perspective from measuring agile potential to current agility. Sidky also used expert assessment and observations instead of distributing a survey to every team member and base the analysis on their overall score. Also, as stated in Paper 1, there might be some fundamental issues with trying to develop an agile measurement model

for all abstraction levels of an organization without also assessing the context of the specific organization. This difficulty is also a reason why we continued our research by breaking down "agility" into smaller subparts, e.g. only investigating the dynamics of what is meant by an agile team in comparison to other types of team.

- Internal validity — **very low**, and as stated in Paper 1, the confounding factors that effect agile maturity models need to be researched a lot more.
- External validity — **not applicable**, since we conducted a validation study of structural construct validity and internal consistency only.

Paper 2 (Chapter 3) is a study on the connections between team maturity and agility, that comprises both qualitative (through interviews) and quantitative data (through a survey). Different types of data have different validity and below is an assessment of the overall validity and the construct validity of the measurements used.

- Reliability — **moderate**
  1. Stability — **low** since one of measurements used were thoroughly tested for stability (the group development questionnaire [94]) and the other was not at all (the agile adoption framework with new categories of items, see Paper 1).
  2. Internal consistency — **high**, the survey can be distributed again and is based on the new categories of questions suggested in Paper 1, and the thoroughly validated group development questionnaire [94].
- Construct Validity — **high**
  1. Consequential — **very high** since practitioners stated the importance of team maturity aspects when building teams. If the results are inaccurate, trying to build a team well in an agile context will probably not, at least, cause harm to the development.
  2. Content — **high** since the two measures used were repeatedly validated by practitioners regarding their applicability in context [94, 130].
  3. Substantive — **low** since one of the measures has a strong theoretical foundation (the group development questionnaire [84]) and the other has shown many issues in regards to what construct it aims at measuring exactly (the agile adoption framework, see Paper 1).
  4. Structural — **very high** since both measures show satisfactory through factor analyses.
  5. External — **moderate**
     (a) Ecological — **very high** since the major part of the study was based on interview data with practitioners.
     (b) Convergent — **not tested**
     (c) Discriminant — **not tested**

(d) Predictive — **high** since we found strong correlations between the agile measurement and the thoroughly validated group maturity measurement, i.e. we can significantly predict variance in group maturity on Scale 4 (Work and productivity) by using the agile measurement.

6. Generalizability — **high** since the group development questionnaire has been used for many different types of human groups and the agility measure was validated by agile experts and also measured qualitatively using interview data.

- Conclusion validity — **high** since the study includes a deeper qualitative analysis and, therefore, an increased possibility to understand the behavior under investigation and the relationships between the constructs. However, it is not very high since only one author thematically analyzed that transcripts. It is also lower because no inter-rater agreement was included.

- Internal validity — **low** since the scarce validation done on the quantitative agility measure is troublesome concerning what that data represents.

- External validity — **moderate** since the interview data provided rich information on how and where the constructs surface and was collected some a range of different companies, however, the population we can generalize to is IT companies in the US and possibly other parts of the world with similar corporate culture. The quantitative survey data sample is too small for broader generalization ($N = 66$).

Paper 3 (Chapter 4) only used validated scales developed by other researchers on substantial datasets to describe the links between agile practices, interpersonal conflict, and perceived productivity. We only had quantitative and cross-sectional data and can, therefore, discuss causality, and our conclusions are dependent on us having a random sample of the intended population of agile teams.

- Reliability — **high**

  1. Stability — **moderate** since the group development questionnaire has been thoroughly tested for stability [94], but the perceptive agile measurement has not.
  2. Internal consistency — **very high**

- Construct Validity — **high**

  1. Consequential — **low** since we do not know the effects of misinterpreting level of conflict or the use of agile practices. However, we assess the severity of such misinterpretation as high since score meanings would severely damage teams if they, for example, have high levels of conflict but obtain low scores and are therefore reluctant to resolute them.
  2. Content — **very high** because both measurements have been developed and tested by experts.

3. Substantive — **very high** since both measures have a strong theoretical foundation and, as compare to Paper 2, the agility measure focuses on the social-psychological effect of agile practices, which clarifies the construct to a large extent.
4. Structural — **very high** since both measures were tested with empirical data and analyzed using statistical factor analysis.
5. External — **very low**
   (a) Ecological — **not tested**, i.e. we did not collect any contextual information about the relationships between the measured constructs.
   (b) Convergent — **not tested**
   (c) Discriminant — **not tested**
   (d) Predictive — **not tested**
6. Generalizability — **high** since the group development questionnaire has been tested with huge sets of data (see, e.g. [75], however, mostly with data from people in the US [94]) and the perceptive agile measurement also has validation conducted on, at least, more data than other agility measures. In contrast, the perceptive agile measurement was validated with data from many different parts of the world [40].

- Conclusion validity — **high** since we used validated measurements and a well-known statistical method.
- Internal validity — **very low** since we have no information about causality or how the constructs under study interact nor do we have any idea about possible confounding factors.
- External validity — **low** since we can only generalize to the population of software development companies in Sweden. We believe similar patterns would exist in other parts of the world, especially since interpersonal conflict is a basic human trait. However, the behavior expressing conflict might differ.

Paper 4 (Chapter 5) investigates the connections between group maturity (again, measured by the group development questionnaire), software development velocity (measured by hours spent on scrum tasks), and planning effectiveness (measured by earned over planned points per sprint).

- Reliability — **very high**
  1. Stability — **very high** since both the group development questionnaire and the external measures of performance can be retaken in the same way.
  2. Internal consistency — **very high** of the same reasons as 1).
- Construct Validity — **high**
  1. Consequential — **high** since the potential harm of misinterpretation is low because working on building a good team cannot be destructive in relation to agile software development.

2. Content — **moderate** since the group development questionnaire scored high in this category, but the other two measurements are proxies for performance, which need to research or perspectives before they can be assessed as equally high. Specifically the measurement of velocity and to compare velocities between teams is a thorny issue.
3. Substantive — **high** but not very high due to the issue of measuring software development velocity.
4. Structural — **very high** since factor analyses were conducted on the group development questionnaire and the performance measurements were external quantitative measures.
5. External — **high**
   (a) Ecological — **very high** since both the quantitative and qualitative data were verified with key participants.
   (b) Convergent — **moderate** if seen from the perspective of investigating the convergent validity of the group development questionnaire. Since it showed significant correlations to a measurement of software development performance, it can be argued to be a valid measurement of temporal team dynamics also in the agile team context.
   (c) Discriminant — **moderate** since we did not test the group development questionnaire against a measure that it theoretically should be dissimilar to. However, the fact that the team maturity level was not related to velocity propose an interesting new finding in relation to this category.
   (d) Predictive — **very high** since, again, the team maturity should be able to predict the performance of any teamwork. However, in hindsight, implementing features fast was more dependent on individual technical skills, which make us conclude that the predictive power the of group development questionnaire is very high.
6. Generalizability — **very low** since we had a unique opportunity to collect similar data from four different teams, we do not know how to collect more data, and more data is needed, to assess the generalizability of the findings.

- Conclusion validity — **very high** since we discussed both the group development issues and the external measurements of performance in interviews with practitioners.
- Internal validity — **very low** since we did not investigate causality between construct at all in any of the data analyses conducted, i.e. we do not have any information of the direction of the influence between planning effectiveness, group maturity, and velocity.
- External validity — **low** since we believe the study shows connections between the studied construct that would be similar outside the specific case. However, as stated above, we need more data from more contexts

and high samples with comparable measures of performance to increase the external validity.

Paper 5 (Chapter 6) is a study on the connections between individual nontechnical skills and the maturity of agile practices in teams. It can be seen as a study of discriminant validity since we investigated if another level of analysis could explain more variance than our group-level constructs.

- Reliability — **low**
  1. Stability — **very low** since the stability was not investigated for the measure of individual skills.
  2. Internal consistency — **low** no such studies or data have been collected for the skills measurement.
- Construct Validity — **moderate**
  1. Consequential — **high** since we do not believe assessing skills nor agile team practices would harm the teams if misinterpreted. However, the issues with measuring self-assessed skills might cause our conclusions to be wrong which in turn might harm companies if they mistakenly change their recruitment and team composition practices.
  2. Content — **moderate** due to the untested, however intuitive, measure of skills.
  3. Substantive — **low** since skills suggested in the related work of software engineering research is mostly based on smaller studies with old and small datasets.
  4. Structural — **low** again due to the skills measurement.
  5. External — **moderate**
     (a) Ecological — **very low** since we do not know the actual skills or how they were assessed in context. In addition, the measurement of individual skills is not a validated procedure.
     (b) Convergent — **not tested**
     (c) Discriminant — **high** since we also checked if the individual nontechnical skills could predict perceived code quality instead in the same study, which it could not either.
     (d) Predictive — **very high** seen from the perspective of the agility measurement since team-level capacity should not be predicted by individual skills. Such a conclusion is new to the software engineering research field, however, not to psychology.
  6. Generalizability — **very low** since we have no information in relation to different groups, settings, or tasks and our way of measuring skills.
- Conclusion validity — **low** because of a large threat to lies in our self-assessment of individual nontechnical skills. Since we only measured perceived skills we do not know if these are connected to real skills,

however, we speculate that the probably are since related studies in psychology suggest people systematically overestimate their skills, which then still makes associative studies possible (i.e. the error is systematic and not random).

- Internal validity — **low** since we do not know if our items capture the actual skills since they have not been used or validated before, and, therefore, there could be a diversity of confounding factors that explain the results.
- External validity — **high** despite a convenience sample of IT departments of different organizations since the sample consists of data from seven organizations on different continents as well as from the public and private sectors. The organizations were also of different sizes, and the participating teams were at different maturity levels, according to the 'gatekeepers' at each company. We also decided to omit role information from the agile team members participating in the research, which would, hopefully, capture the agile practices from a team perspective and not just from the perspective of software developers. In summary, we believe the sample reflects agile team practices, at least in large parts of South America and Europe.

Summary We will finish this section by summarizing the validity overall objective of this thesis, i.e. the relationship between group development (team maturity) and what is meant by an agile team. To state if group maturity is a prerequisite for agility, more and larger studies are needed and preferably longitudinal ones, to draw conclusions about causality. The construct validity of 'agility' is probably the largest threat to the papers that investigate 'team agility' in this thesis. However, we contribute to defining the subcategory of 'agile teams' as mature (Stage 4) workgroups as described in social psychology to at least pinpoint what agility means on a team level. The correlation shown in Paper 2 (Chapter 3) could be seen to strengthen the agile maturity tool since it had significant concurrent validity to another validated tool. If agility means at least some aspects of group maturity, the maturity model indeed captured some aspects of it, since the measurement was correlated to the group development questionnaire (GDQ). Also, we could have internal validity threats in the form of other uncontrolled variables affecting both the agility and group maturity measurements, but the interviews mitigated some of these threats to the survey design. The reliability of qualitative approaches is usually considered lower than collecting quantitative data. When it comes to the interviews etc. that we conducted, the perception of the researcher introduced social bias, which is a threat to conclusion validity. This threat means that these studies are less reliable since they are difficult to replicate. However, the combination of validated quantitative surveys, external project performance data, and qualitative interviews triangulates the issue of group development and agility, which then mitigates and increases the reliability and validity overall. We also have data from other abstraction levels of the organization to support our hypotheses.

On a final remark, we would like to state that analyzing and navigating through validity threats throughout a research program is a very challenging

endeavor. We think that our proposed checklist was useful and made us realize what validation studies we need to conduct in the future to strengthen the validity of the conclusions reached. As also mentioned, we think it is imperative to change the culture in software engineering research from seeing tool-constructing as the holy grail of research and instead value validation studies higher. Such studies are crucial to theory building and, just like in psychological research, spending an entire Ph.D. candidacy on the validation of one single measurement tool should be, not only approved but encouraged.

1.7 Conclusions and future work

This thesis set out to investigate how group maturity is related to team agility in software development. Through a set of studies employing different research designs, we have found that agile work groups at different group development stages will adopt team agility differently. These findings are important contributions to both researchers and practitioners since they introduce an important temporal perspective to building agile teams that need to be taken into consideration in both research and practice to understand the dynamics of agile teams. We have specifically focused on group development in connection with agile teams, but have also conducted a validation study of agility to understand the construct better. In addition, we changed the level of analysis and validated our assumption of using the work-group as the level of analysis by both investigating strategic agility, and if individual nontechnical skills could explain the mature use of agile practices.

The most natural next step is to investigate the agile practices explicitly, as defined by So et al. [40], and the whole group development questionnaire. The measurement of Scale 4 has been shown to correlate to effectiveness measurements in other fields. A couple of examples are; ability to finish projects faster [131], better student performance on a standardized test (SAT scores) if faculty team is mature [132, 133], and lower mortality rates in intensive care units performing surgery [134]. Future research could also collect more data from different parts of the world and see if group maturity and agility differ depending on culture. We mostly employed a cross-sectional research design, which gives us difficulty in drawing any conclusions about causal relationships. Future research should research these concepts over time in longitudinal studies. For example, 'could we increase agility by helping the team in its group development?' Or, 'does an agile software development process lead to the group maturing?', which Schölkopf and his colleagues' latest work could help us answer (see, e.g. [135]).

Social identity theory could be utterly useful when navigating through the added complexity of the different social relationship surfacing in an agile project. However, the theory could be seen as complicated and hard to grasp for people without a behavioral science background, which means researchers must first run experiments to gather empirical evidence to build a theory of 'agility' eventually, and then provide scientifically founded and validated guidelines to practitioners.

Another direction for practice could be the application and realization of an aspect that Lewin [136] already suggested in the 1930s. He showed that it is

possible to link a variety of facts of individual and social psychology when, instead of using classification, describing them in terms of how they affect the situation. Examples from the software engineering context might be the process of learning and orientation, time perspective, planning, problems of individual maturation, technical skills, conflicts and tension, and team cohesion. The concept of 'agility' has been shown to be highly contextual, which is natural since achieving agility is something existing in the 'force fields' within different types of organization (as expressed in Lewinian vocabulary). Lewin [136] suggested replacing classifications with the construction, derivation, and axiomatization of laws, meaning that the inter-dependencies of objects and events are only relevant in relation to the systems they are in. This approach can be directly applied to achieving 'agility' since the agile principles can be viewed as the end-goal and all the forces (from all different abstraction levels of the organization) can be understood in relation to their effects on the specific system (i.e. the organizational context).

Finally, a more long-term future step is to introduce university courses in behavioral software engineering [17] that comprise much-needed training for students in social-psychological factors influencing software development projects.

Chapter 2

Paper 1

The prospects of a quantitative measurement of agility: A validation study on an agile maturity model

L. Gren, R. Torkar, R. Feldt

Journal of Systems and Software, 107, 38–49, 2015.

Chapter 3

Paper 2

Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies

L. Gren, R. Torkar, R. Feldt

Journal of Systems and Software, 124, 104–119, 2017.

Chapter 4

Paper 3

The links between agile practices, interpersonal conflict, and perceived productivity

L. Gren

Proceedings of the 21st Conference on Evaluation and Assessment in Software Engineering (EASE), Karlskrona, Sweden, June 15–16, 292–297, 2017.

Chapter 5

Paper 4

The connections between group maturity, software development velocity and planning effectiveness

K. Al-Sabbagh, L. Gren

The Journal of Software: Evolution and Process (Accepted August 8, 2017).

Chapter 6

Paper 5

Non-technical individual skills are weakly connected to the maturity of agile practices

L. Gren, A. Knauss, C. Stettina

In submission to journal.

References

[1] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," *The Journal of Systems and Software*, vol. 85, pp. 1213–1221, 2012.

[2] S. P. Philbin, "Investigating the scope for agile project management to be adopted by higher education institutions," in *Software Project Management for Distributed Computing*. Springer, 2017, pp. 345–366.

[3] J. Pajares, D. Poza, F. Villafañez, and A. López-Paredes, "Project management methodologies in the fourth technological revolution," in *Advances in Management Engineering*. Springer, 2017, pp. 121–144.

[4] L. Williams, "What agile teams think of agile principles," *Communications of the ACM*, vol. 55, no. 4, pp. 71–76, 2012.

[5] M. Engwall, "The futile dream of the perfect goal," in *Beyond Project Management*, K. Sahlin-Andersson and A. Söderholm, Eds. Malmö: Liber Abstrakt, 2002, pp. 3261–3277.

[6] K. Petersen, "Is lean agile and agile lean? A comparison between two software development paradigms," in *Modern software engineering concepts and practices: Advanced approaches*, A. H. Dogru and V. Bicer, Eds. Hershey, PA, US: IGI Global, 2011, pp. 19–46.

[7] C. Cobb, *Making Sense of Agile Project Management: Balancing Control and Agility*. Hoboken: John Wiley & Sons, Inc., 2011.

[8] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and software technology*, vol. 50, no. 9, pp. 833–859, 2008.

[9] G. Melnik and F. Maurer, "Comparative analysis of job satisfaction in agile and non-agile software development teams," in *Extreme Programming and Agile Processes in Software Engineering*. Springer, 2006, pp. 32–42.

[10] P. Serrador and J. K. Pinto, "Does agile work? – A quantitative analysis of agile project success," *International Journal of Project Management*, vol. 33, no. 5, pp. 1040–1051, 2015.

[11] P. Lenberg, R. Feldt, and L.-G. Wallgren, "Human factors related challenges in software engineering: An industrial perspective," in *Proceedings of the Eighth International Workshop on Cooperative and Human Aspects of Software Engineering*. IEEE, 2015, pp. 43–49.

[12] E. Whitworth and R. Biddle, "The social nature of agile teams," in *Agile Conference (AGILE)*. IEEE, 2007, pp. 26–36.

[13] J. Iivari and N. Iivari, "The relationship between organizational culture and the deployment of agile methods," *Information and Software Technology*, vol. 53, no. 5, pp. 509–520, 2011.

[14] C. Tolfo and R. Wazlawick, "The influence of organizational culture on the adoption of extreme programming," *Journal of systems and software*, vol. 81, no. 11, pp. 1955–1967, 2008.

[15] E. Schein, *Organizational culture and leadership*, 4th ed. San Francisco: Jossey-Bass, 2010.

[16] C. Tolfo, R. Wazlawick, M. Ferreira, and F. Forcellini, "Agile methods and organizational culture: Reflections about cultural levels," *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 23, no. 6, pp. 423–441, 2011.

[17] P. Lenberg, R. Feldt, and L.-G. Wallgren, "Behavioral software engineering: A definition and systematic literature review," *Journal of Systems and Software*, vol. 107, pp. 15 – 37, 2015.

[18] S. W. Kozlowski and D. R. Ilgen, "Enhancing the effectiveness of work groups and teams," *Psychological science in the public interest*, vol. 7, no. 3, pp. 77–124, 2006.

[19] J. R. Hackman, "Learning more by crossing levels: Evidence from airplanes, hospitals, and orchestras," *Journal of organizational behavior*, vol. 24, no. 8, pp. 905–922, 2003.

[20] K. Beck, "Embracing change with extreme programming," *IEEE Computer*, vol. 32, no. 10, pp. 70–77, 1999.

[21] A. Cockburn and J. Highsmith, "Agile software development: The people factor," *IEEE Computer*, no. 11, pp. 131–133, 2001.

[22] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Computers in Human Behavior*, vol. 51, pp. 915–929, 2015.

[23] R. Hoda, J. Noble, and S. Marshall, "Negotiating contracts for agile projects: A practical perspective," in *International Conference on Agile Processes and Extreme Programming in Software Engineering*. Springer, 2009, pp. 186–191.

[24] M. O. Ahmad, J. Markkula, and M. Oivo, "Kanban in software development: A systematic literature review," in *39th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2013, pp. 9–16.

[25] L. Rising and N. S. Janoff, "The scrum software development process for small teams," *IEEE software*, vol. 17, no. 4, pp. 26–32, 2000.

[26] S. A. Licorish, J. Holvitie, S. Hyrynsalmi, V. Leppnen, R. O. Spnola, T. S. Mendes, S. G. MacDonell, and J. Buchan, "Adoption and suitability of software development methods and practices," in *23rd Asia-Pacific Software Engineering Conference (APSEC)*, December 2016, pp. 369–372.

[27] S. Robson, *Agile SAP: Introducing Flexibility, Transparency and Speed to SAP Implementations*. Cambridgeshire: IT Governance Publishing, 2013.

[28] J. Cooke, *Agile: An executive Guide. Real results from IT budgets*. Cambridgeshire: IT Governance Publishing, 2011.

[29] K. Schwaber, *Agile project management with Scrum*. Redmond, Wash.: Microsoft Press, 2004.

[30] G. Weinberg, *The psychology of computer programming*, silver anniversary ed. ed. New York: Dorset House Pub, 1998.

[31] G. Kim, K. Behr, and G. Spafford, *The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win*. IT Revolution Press, 2013.

[32] G. Chin, *Agile project management: How to succeed in the face of changing project requirements*. New York: AMACOM, 2004.

[33] A. Cockburn, *Agile software development: The cooperative game*, 2nd ed. Upper Saddle River, NJ: Addison-Wesley, 2007.

[34] M. van Oosterhout, *Business Agility and Information Technology in Service Organizations*. Rotterdam: Erasmus Research Institute of Management, 2010.

[35] R. Hoda, N. Salleh, J. Grundy, and H. M. Tee, "Systematic literature reviews in agile software development: A tertiary study," *Information and Software Technology*, vol. 85, pp. 60–70, 2017.

[36] T. Tapanainen, M. Hallanoro, J. Päivärinta, and H. Salmela, "Towards an agile IT organisation: A review of prior literature," in *Proceedings of the 2nd European Conference on Information Management and Evaluation*, 2008, pp. 425–432.

[37] M. Hummel, C. Rosenkranz, and R. Holten, "The role of communication in agile systems development," *Business & Information Systems Engineering*, vol. 5, no. 5, pp. 343–355, 2013.

[38] C. d. O. Melo, C. Santana, and F. Kon, "Developers motivation in agile teams," in *38th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2012, pp. 376–383.

[39] M. Wufka and P. Ralph, "Explaining agility with a process theory of change," in *Agile Conference (AGILE)*. IEEE, 2015, pp. 60–64.

[40] C. So and W. Scholl, "Perceptive agile measurement: New instruments for quantitative studies in the pursuit of the social-psychological effect of agile practices," in *Agile Processes in Software Engineering and Extreme Programming*. Springer, 2009, pp. 83–93.

[41] C. So, *Making software teams effective: How agile practices lead to project success through teamwork mechanisms*. Frankfurt am Main: Peter Lang, 2010.

[42] J. Keyton, *Communicating in groups: Building relationships for group effectiveness*. New York: McGraw-Hill, 2002.

[43] M. A. Hogg and G. M. Vaughan, *Social Psychology*, 7th ed. Harlow, England: Pearson, 2014.

[44] P. Jalote, A. Palit, and P. Kurien, "The timeboxing process model for iterative software development," ser. Advances in Computers. Elsevier, 2004, vol. 62, pp. 67 – 103.

[45] J. Combs, Y. Liu, A. Hall, and D. Ketchen, "How much do high-performance work practices matter? A meta-analysis of their effects on organizational performance," *Personnel psychology*, vol. 59, no. 3, pp. 501–528, 2006.

[46] W. R. Evans and W. D. Davis, "High-performance work systems and organizational performance: The mediating role of internal social structure," *Journal of management*, vol. 31, no. 5, pp. 758–775, 2005.

[47] D. Wat and M. A. Shaffer, "Equity and relationship quality influences on organizational citizenship behaviors: The mediating role of trust in the supervisor and empowerment," *Personnel review*, vol. 34, no. 4, pp. 406–422, 2005.

[48] O. McHugh, K. Conboy, and M. Lang, "Agile practices: The impact on trust in software project teams," *IEEE Software*, vol. 29, no. 3, pp. 71–76, 2012.

[49] D. A. Buchanan, "You stab my back, I'll stab yours: Management experience and perceptions of organization political behaviour," *British Journal of Management*, vol. 19, no. 1, pp. 49–64, 2008.

[50] B. M. Bass, "From transactional to transformational leadership: Learning to share the vision," *Organizational dynamics*, vol. 18, no. 3, pp. 19–31, 1990.

[51] L. Karlsson, T. Thelin, B. Regnell, P. Berander, and C. Wohlin, "Pair-wise comparisons versus planning game partitioning – Experiments on requirements prioritisation techniques," *Empirical Software Engineering*, vol. 12, no. 1, pp. 3–33, 2007.

[52] V. Mahnič and T. Hovelja, "On using planning poker for estimating user stories," *Journal of Systems and Software*, vol. 85, no. 9, pp. 2086–2095, 2012.

[53] B. J. Hicks, "Lean information management: Understanding and eliminating waste," *International journal of information management*, vol. 27, no. 4, pp. 233–249, 2007.

[54] J. M. Levine and R. L. Moreland, "Group socialization: Theory and research," *European review of social psychology*, vol. 5, no. 1, pp. 305–336, 1994.

[55] V. Stray, D. I. Sjøberg, and T. Dybå, "The daily stand-up meeting: A grounded theory study," *Journal of Systems and Software*, vol. 114, pp. 101–124, 2016.

[56] M. Hewstone, M. Rubin, and H. Willis, "Intergroup bias," *Annual review of psychology*, vol. 53, no. 1, pp. 575–604, 2002.

[57] J. J. van Bavel and W. A. Cunningham, "A social neuroscience approach to self and social categorisation: A new look at an old issue," *European Review of Social Psychology*, vol. 21, no. 1, pp. 237–284, 2010.

[58] M. A. Hogg and K. D. Williams, "From I to we: Social identity and the collective self," *Group Dynamics: Theory, Research, and Practice*, vol. 4, no. 1, p. 81, 2000.

[59] H. Tajfel, M. G. Billig, R. P. Bundy, and C. Flament, "Social categorization and intergroup behaviour," *European journal of social psychology*, vol. 1, no. 2, pp. 149–178, 1971.

[60] D. R. Denison, S. L. Hart, and J. A. Kahn, "From chimneys to cross-functional teams: Developing and validating a diagnostic model," *Academy of management journal*, vol. 39, no. 4, pp. 1005–1023, 1996.

[61] K. Petersen, C. Wohlin, and D. Baca, "The waterfall model in large-scale development," in *International Conference on Product-Focused Software Process Improvement*. Springer, 2009, pp. 386–400.

[62] N. Moe, A. Aurum, and T. Dybå, "A teamwork model for understanding an agile team: A case study of a scrum project," *Information and Software Technology*, vol. 52, no. 5, pp. 480–491, 2010.

[63] E. Derby and D. Larsen, *Agile retrospectives: Making good teams great*. Raleigh, NC: Pragmatic Bookshelf, 2006.

[64] S. I. Tannenbaum and C. P. Cerasoli, "Do team and individual debriefs enhance performance? A meta-analysis," *Human factors*, vol. 55, no. 1, pp. 231–245, 2013.

[65] T. O. Lehtinen, J. Itkonen, and C. Lassenius, "Recurring opinions or productive improvements — What agile teams actually discuss in retrospectives," *Empirical Software Engineering*, vol. 22, no. 5, pp. 2409–2452, 2017.

[66] W. Krebs, "Turning the knobs: A coaching pattern for XP through agile metrics," in *Conference on Extreme Programming and Agile Methods*. Springer, 2002, pp. 60–69.

[67] J. F. Dovidio, S. L. Gaertner, and A. Validzic, "Intergroup bias: Status, differentiation, and a common in-group identity." *Journal of personality and social psychology*, vol. 75, no. 1, p. 109, 1998.

[68] D. E. Strode, S. L. Huff, B. Hope, and S. Link, "Coordination in co-located agile software development projects," *Journal of Systems and Software*, vol. 85, no. 6, pp. 1222–1238, 2012.

[69] S. Berczuk, "Back to basics: The role of agile principles in success with a distributed scrum team," in *Agile Conference (AGILE), 2007*. IEEE, 2007, pp. 382–388.

[70] J. Noll, S. Beecham, and I. Richardson, "Global software development and collaboration: Barriers and solutions," *ACM Inroads*, vol. 1, no. 3, pp. 66–78, 2010.

[71] A. W. Woolley, C. F. Chabris, A. Pentland, N. Hashmi, and T. W. Malone, "Evidence for a collective intelligence factor in the performance of human groups," *Science*, vol. 330, no. 6004, pp. 686–688, 2010.

[72] D. Engel, A. W. Woolley, L. X. Jing, C. F. Chabris, and T. W. Malone, "Reading the mind in the eyes or reading between the lines? Theory of mind predicts collective intelligence equally well online and face-to-face," *PloS ONE*, vol. 9, no. 12, pp. 1–16, 2014.

[73] M. A. Hogg and D. I. Terry, "Social identity and self-categorization processes in organizational contexts," *Academy of management review*, vol. 25, no. 1, pp. 121–140, 2000.

[74] B. Tuckman and M. Jensen, "Stages of small-group development revisited," *Group & Organization Management*, vol. 2, no. 4, pp. 419–427, 1977.

[75] S. Wheelan, B. Davidson, and F. Tilin, "Group development across time: Reality or illusion?" *Small group research*, vol. 34, no. 2, pp. 223–245, 2003.

[76] T. Chow and D.-B. Cao, "A survey study of critical success factors in agile software projects," *Journal of Systems and Software*, vol. 81, no. 6, pp. 961–971, 2008.

[77] S. Wheelan, *Creating effective teams: A guide for members and leaders*, 4th ed. Thousand Oaks: SAGE, 2013.

[78] S. McDonald and H. Edwards, "Who should test whom?" *Communications of the ACM*, vol. 50, no. 1, pp. 66–71, 2007.

[79] T. Seger, O. Hazzan, and R. Bar-Nahor, "Agile orientation and psychological needs, self-efficacy, and perceived support: A two job-level comparison," in *Agile Conference (AGILE)*. IEEE, 2008, pp. 3–14.

[80] V. Balijepally, R. Mahapatra, and S. Nerur, "Assessing personality profiles of software developers in agile development teams," *Communications of the Association for Information Systems*, vol. 18, no. 1, p. 4, 2006.

[81] R. Feldt, L. Angelis, R. Torkar, and M. Samuelsson, "Links between the personalities, views and attitudes of software engineers," *Information and Software Technology*, vol. 52, no. 6, pp. 611–624, 2010.

[82] S. Cruz, F. da Silva, and L. Capretz, "Forty years of research on personality in software engineering: A mapping study," *Computers in Human Behavior*, vol. 46, pp. 94–113, 2015.

[83] J. E. Hannay, E. Arisholm, H. Engvik, and D. I. Sjoberg, "Effects of personality on pair programming," *IEEE Transactions on Software Engineering*, vol. 36, no. 1, pp. 61–80, 2010.

[84] S. Wheelan, *Group processes: A developmental perspective*, 2nd ed. Boston: Allyn and Bacon, 2005.

[85] G. Melnik and F. Maurer, "Direct verbal communication as a catalyst of agile knowledge sharing," in *Agile Development Conference, 2004*. IEEE, 2004, pp. 21–31.

[86] V. Balijepally, "Collaborative software development in agile methodologies – Perspectives from small group research," in *Proceedings of the 11th Americas Conference on Information Systems*. AISeL, 2005, pp. 3138–3146.

[87] R. Feldt, R. Torkar, L. Angelis, and M. Samuelsson, "Towards individualized software engineering: Empirical studies should collect psychometrics," in *Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*. ACM, 2008, pp. 49–52.

[88] S. Bryant, "XP: Taking the psychology of programming to the eXtreme," in *Psychology of Programming Interest Group Annual Workshop*. Citeseer, 2004.

[89] N. Moe, A. Aurum, and T. Dybå, "Challenges of shared decision-making: A multiple case study of agile software development," *Information and Software Technology*, vol. 54, no. 8, pp. 853–865, 2012.

[90] R. Hoda, J. Noble, and S. Marshall, "Self-organizing roles on agile software development teams," *IEEE Transactions on Software Engineering*, vol. 39, no. 3, pp. 422–444, 2013.

[91] S. Wheelan, "Group size, group development, and group productivity," *Small Group Research*, vol. 40, no. 2, pp. 247–262, 2009.

[92] W. Bion, *Experiences in groups: and other papers.* London: Routledge, 1992.

[93] Y. Agazarian and S. Gantt, "Phases of group development: Systems-centered hypotheses and their implications for research and practice." *Group Dynamics: Theory, Research, and Practice*, vol. 7, no. 3, p. 238, 2003.

[94] S. Wheelan and J. Hochberger, "Validation studies of the group development questionnaire," *Small Group Research*, vol. 27, no. 1, pp. 143–170, 1996.

[95] R. M. Lerner, M. A. Easterbrooks, and J. Mistry, *Handbook of psychology. Vol. 6, Developmental psychology.* New York: Wiley, 2003.

[96] M. Rapp Ricciardi and J. Schaller, *Projektpsykologi: En introduktion [Project psychology: An introduction].* Lund: Studentlitteratur, 2005.

[97] P. Hersey, K. H. Blanchard, and W. E. Natemeyer, "Situational leadership, perception, and the impact of power," *Group & Organization Management*, vol. 4, no. 4, pp. 418–428, 1979.

[98] S. W. Kozlowski, D. J. Watola, J. M. Jensen, B. H. Kim, and I. C. Botero, "Developing adaptive teams: A theory of dynamic team leadership," in *Team effectiveness in complex organizations: Cross-disciplinary perspectives and approaches*, E. Salas, G. F. Goodwin, and C. S. Burke, Eds. New York, US: Routledge, 2009, pp. 113–155.

[99] L. Greiner, "Evolution and revolution as organizations grow," *Harvard Business Review*, vol. 50, no. 4, 1972.

[100] I. Adizes, "Organizational passages — Diagnosing and treating lifecycle problems of organizations," *Organizational dynamics*, vol. 8, no. 1, pp. 3–25, 1979.

[101] F. Zieris and S. Salinger, "Doing scrum rather than being agile: A case study on actual nearshoring practices," in *Global Software Engineering (ICGSE), 2013 IEEE 8th International Conference on*, Aug 2013, pp. 144–153.

[102] F. Laloux, *Reinventing organizations: A guide to creating organizations inspired by the next stage of human consciousness*, 1st ed. Brussels, Belgium: Nelson Parker, 2014.

[103] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2009.

[104] D. S. Gardner, "Perception and memory of witnesses," *Cornell Law Quarterly*, vol. 18, pp. 391–409, 1932.

[105] J. C. Hafele and R. E. Keating, "Around-the-world atomic clocks: Predicted relativistic time gains," *Science*, vol. 177, no. 4044, pp. 166–168, 1972.

[106] J. P. Forgas, "Feeling and doing: Affective influences on interpersonal behavior," *Psychological inquiry*, vol. 13, no. 1, pp. 1–28, 2002.

[107] Z. Kunda, "The case for motivated reasoning," *Psychological bulletin*, vol. 108, no. 3, p. 480, 1990.

[108] D. W. McAllister, T. R. Mitchell, and L. R. Beach, "The contingency model for the selection of decision strategies: An empirical test of the effects of significance, accountability, and reversibility," *Organizational Behavior and Human Performance*, vol. 24, no. 2, pp. 228–244, 1979.

[109] T. Pyszczynski and J. Greenberg, "Toward an integration of cognitive and motivational perspectives on social inference: A biased hypothesis-testing model," *Advances in experimental social psychology*, vol. 20, pp. 297–340, 1987.

[110] Z. Kunda and R. Sanitioso, "Motivated changes in the self-concept," *Journal of Experimental Social Psychology*, vol. 25, no. 3, pp. 272–285, 1989.

[111] M. Ross, C. McFarland, and G. J. Fletcher, "The effect of attitude on the recall of personal histories," *Journal of Personality and Social Psychology*, vol. 40, no. 4, p. 627, 1981.

[112] P. Kitcher, *Living with Darwin: Evolution, design, and the future of faith.* Oxford: Oxford University Press, 2007.

[113] M. Altman, *Handbook of contemporary behavioral economics: Foundations and developments.* Armonk, N.Y.: M.E. Sharpe, 2006.

[114] L. Adkins, *Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition.* Boston, M.A.: Pearson Education, 2010.

[115] R. Turner and A. Jain, "Agile meets cmmi: Culture clash or common cause?" in *Extreme Programming and Agile Methods: XP/Agile Universe 2002.* Springer, 2002, pp. 153–165.

[116] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering.* Berlin: Springer, 2012.

[117] R. Feldt and A. Magazinius, "Validity threats in empirical software engineering research – an initial survey," in *International Conference on Software Engineering and Knowledge Engineering (SEKE)*, 2010, pp. 374–379.

[118] S. Messick, "Validity of psychological assessment: Validation of inferences from persons' responses and performances as scientific inquiry into score meaning," *American psychologist*, vol. 50, no. 9, p. 741, 1995.

[119] ——, "Standards of validity and the validity of standards in performance assessment," *Educational measurement: Issues and practice*, vol. 14, no. 4, pp. 5–8, 1995.

[120] G. F. Kuder and M. W. Richardson, "The theory of the estimation of test reliability," *Psychometrika*, vol. 2, no. 3, pp. 151–160, 1937.

[121] L. Cronbach, "Coefficient alpha and the internal structure of tests," *Psychometrika*, vol. 16, no. 3, pp. 297–334, 1951.

[122] T. Gilovich, "Biased evaluation and persistence in gambling," *Journal of personality and social psychology*, vol. 44, no. 6, p. 1110, 1983.

[123] Z. Ginossar and Y. Trope, "Problem solving in judgment under uncertainty," *Journal of Personality and social Psychology*, vol. 52, no. 3, p. 464, 1987.

[124] D. Kahneman, *Thinking, fast and slow*. New York: Farrar, Straus and Giroux, 2011.

[125] R. S. Nickerson, "Confirmation bias: A ubiquitous phenomenon in many guises," *Review of general psychology*, vol. 2, no. 2, p. 175, 1998.

[126] M. Jørgensen and E. Papatheocharous, "Believing is seeing: Confirmation bias studies in software engineering," in *41st Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2015, pp. 92–95.

[127] M. Jørgensen, T. Dybå, K. Liestøl, and D. I. Sjøberg, "Incorrect results in software engineering experiments: How to improve research practices," *Journal of Systems and Software*, vol. 116, pp. 133–145, 2016.

[128] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, "Active learning increases student performance in science, engineering, and mathematics," *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8410–8415, 2014.

[129] T. J. Polderman, B. Benyamin, C. A. De Leeuw, P. F. Sullivan, A. Van Bochoven, P. M. Visscher, and D. Posthuma, "Meta-analysis of the heritability of human traits based on fifty years of twin studies," *Nature genetics*, vol. 47, no. 7, pp. 702–709, 2015.

[130] A. Sidky, "A structured approach to adopting agile practices: The agile adoption framework," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2007.

[131] S. Wheelan, D. Murphy, E. Tsumura, and S. F. Kline, "Member perceptions of internal group dynamics and productivity," *Small Group Research*, vol. 29, no. 3, pp. 371–393, 1998.

[132] S. Wheelan and F. Tilin, "The relationship between faculty group development and school productivity," *Small group research*, vol. 30, no. 1, pp. 59–81, 1999.

[133] S. Wheelan and J. Kesselring, "Link between faculty group: Development and elementary student performance on standardized tests," *The journal of educational research*, vol. 98, no. 6, pp. 323–330, 2005.

[134] S. Wheelan, C. N. Burchill, and F. Tilin, "The link between teamwork and patients' outcomes in intensive care units," *American Journal of Critical Care*, vol. 12, no. 6, pp. 527–534, 2003.

[135] E. Sgouritsa, D. Janzing, P. Hennig, and B. Schölkopf, "Inference of cause and effect with unsupervised inverse regression," in *Artificial Intelligence and Statistics*, 2015, pp. 847–855.

[136] K. Lewin, "Field theory and experiment in social psychology: Concepts and methods," *American journal of sociology*, vol. 44, no. 6, pp. 868–896, 1939.