Thesis for The Degree of Licentiate of Engineering

Using the Work and Organizational Psychology Perspective in Research on Agile Software Development Teams

Lucas Gren

Division of Software Engineering
Department of Computer Science & Engineering
Chalmers and University of Gothenburg
Gothenburg, Sweden, 2015

Using the Work and Organizational Psychology Perspective in Research on Agile Software Development Teams

Lucas Gren

Technical Report No 142L
ISSN 1652-876X
Department of Computer Science & Engineering
Division of Software Engineering
Chalmers University of Technology and the University of Gothenburg
Gothenburg, Sweden

This thesis was typeset with LaTeX.
Printed by Chalmers Reproservice,
Gothenburg, Sweden 2015.

"All models are wrong. Some are useful." – George E.P. Box

Abstract

Background The development of software has gone from more strict plan-driven projects to involve more human interaction and communication due to approaches like agile software development. With the realization of the importance of psychological aspect comes the possibility of learning from other more established research fields instead of reinventing the wheel.

Objective In the field of work and organizational psychology there is an extensive body of knowledge of work-life in many different contexts. The objective of this thesis is to show some examples of how both methods and models from psychology research can be used in software engineering and specifically to understand agile software development teams. The selected models and tools were; new aspects of work motivation in agile teams in larger organizations, statistical tests of validation (factor analysis), and using the social psychology model of group development in connection to agile teams.

Method The appended papers consist of both exploratory, correlative and validation studies. The research methods range from interviews, focus groups, and survey data as well as qualitative and quantitative interpretations. Eight companies participated consisting of two European-based and six US-based organizations, and a total of 76 people participated in the studies. The data collection procedures were also diverse ranging from recorded in-person interviews and focus groups, to online surveys and remotely recorded phone interviews.

Results The analysis included thematic ditto of interview transcripts, correlation of variables in survey data, and statistical validation tests of a survey itself. Some studies used one research methodology while other triangulate the research question in order to increase the validity of the results. The results strongly indicate that many agile maturity models need more validation, that there are work motivational aspects of employees working on agile teams in a more traditional structure, and that the group development aspect of building agile teams contributes with concrete guidance on moving teams forward.

Conclusions We conclude that there are a set of useful methods and models in work and organizational psychology that are applicable, specifically, to the agile software development context of teams, but also, more generally to a larger perspective of software engineering that involves human factors. This thesis will hopefully convince researchers and practitioners of the usefulness of adding the psychological dimension when trying to understand such social and complex systems.

Keywords

Software Engineering, Work and Organizational Psychology, Agile Software Development, Empirical Research, Group Development

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 main supervisor Professor Richard Torkar, my co-supervisor Doctor Richard Berntsson Svensson, and Professor Robert Feldt. I would also like to thank my love Maja, my family, and all my friends who all put up with me and my obsessions of science and other things in life.

List of Publications

Appended papers

This thesis is based on the following papers:

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), 2014, Orlando, Florida, July 28-August 1, 2014 pp. 11–15).

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

3. 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'2015), Madeira, Portugal, August 26-28, 2015 pp. 1–8.

4. L. Gren, R. Torkar, R. Feldt "Connections between group maturity and agility: A quantitative and qualitative investigation at eight companies"
In Submission to Journal.

Papers not included in the thesis

The following papers are not included in the thesis:

1. 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'2015), Madeira, Portugal, August 26-28, 2015 pp. 104–111.

2. C. Konstantinos, L. Gren "Agility measurements mismatch: A validation study on three agile team assessments in software engineering"
In Submission to Conference.

Contents

Chapter 1

Introduction

Psychological aspects of software engineering and the importance of human factors to build high-quality software fast, have gotten more attention recently [1, 2]. The field of psychology and its sub-field of work and organizational psychology have spent decades researching human endeavor in the workplace [3]. Not only does the field have models and methods applicable to the software engineering (SE) field, but also a well-developed scientific and empirical approach to social science research [4].

Within the larger field of work and organizational psychology many research, and practical, findings are applicable to the context of software engineering. Work and organizational psychology studies human endeavor in the workplace on a higher abstraction level than software engineering but includes countless studies on human communication and cooperation. New ways of working (or development processes in the SE context) can be seen as examples of an organizational change (or organizational development) since we try to implement a new structure and ways of working in an organization of people. Therefore, there is an extensive body of knowledge directly applicable to the transition to more recent development techniques, like agile software development. Some of these aspects have, of course, been used in software engineering, but some other areas of concern have yet to reach the research field. That is why this thesis does not only include studies on applicable psychological perspectives and methods applied in the software engineering context, but also statistical validation techniques (used in psychology research) applied to methods already developed in software engineering.

There are some studies that try to connect psychology, cultural aspects, and/or group development aspects to software engineering. These are described in the Introduction Section of Paper 4 (Chapter 5). In order for the reader to fully understand the content of this thesis, a basic understanding of agile software development and its underlying principles is needed. A description of agility and a comparison to traditional project management can be found in the Introduction Section of Paper 1 (Chapter 2). When it comes to measuring agility, a background to agility research in connection to agile maturity models can be found in the Related Work Section in Paper 2 (Chapter 3). Also, the reader needs to comprehend the ideas behind building effective teams and group developmental psychology. For a description of these concepts see

Related Work Section in Paper 4 (Chapter 5).

In order to fill some of the research gaps in applying work and organizational psychology concepts to software engineering, this thesis uses ideas from psychology applied to the agile software development context seen though the lens of organizational development. The main approaches to fill some of the gaps are summarized as follows:

1. A new perspective of work motivation in the agile context (Paper 1).

2. A statistical validation technique often used in psychology research applied to an agile maturity survey (Paper 2).

3. An analysis of the connection between group maturity (a social psychology theory) and agility (Papers 3 and 4).

These studies use both qualitative and quantitative data in form of interviews, focus groups, and surveys to explore how techniques and aspects from work and organizational psychology could be used to increase our understanding of agile software development.

In the next section (Section 1.1) we will present the research focus for this thesis including the research objective and the research questions. After that, in Section 1.2, we will extend the theoretical background given by the papers in order to answer these questions. 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 short summary of each papers and their contribution. Section 1.5 will discuss the paper contributions in connection the the theoretical background in order to fulfill the overall research objective and present limitations to this thesis. Finally Section 1.6 will give conclusions and suggest future work.

1.1 Research focus

The goal of this thesis is to see if one can use the perspective of work and organizational psychology to deepen the understanding of the aspects of agile software development that include human factors. This is, of course, a broad subject, and includes both the actual models found in the research field of psychology applied in the software engineering context, but also an application of statistical tests used in psychology for more than half a century. Some psychological aspects have been somewhat researched in software engineering and these are described in the Related Work Section in Paper 4 (Chapter 5). Work and organizational psychology provides many different applicable methods and models. Many of the ones applicable to a transition to agile software development are provided in the sub-field of organizational change and (for cultural aspects of change) in the theory of organizational development (OD). Figure 1.1 clarifies how these concepts are connected in this thesis.

The research objective is therefore to provide ways of using work and organizational psychology to increase our understanding of agile software development. The research questions from each paper help to find an answer to the this research objective. These are:

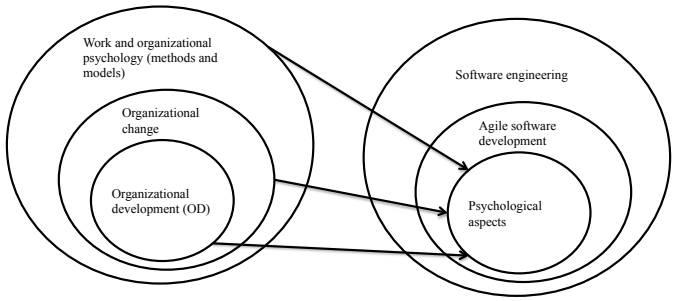

Figure 1.1: Clarification of related concepts used in this thesis.

- RQ1: How can we use work motivational psychology to increase our understanding of agile teams in a non-agile surrounding organization? (Paper 1)
- RQ2: How can we use quantitative statistical tests (used in psychology research) to validate surveys of agile maturity? (Paper 2)
- RQ3: How can we apply group developmental theory to increase our understanding of agile teams? (Paper 3 and 4)

We will now give a short overview of agile software development in order to clarify its connections to work and organizational psychology. All of these aspect are needed as a complement to the related work written in the actual papers, to understand the overall research strategy.

1.2 Theoretical Background

This theoretical background is intended as a complement to the theory given in the related work sections of each paper. In order to give the reader a deeper understanding of how work and organizational psychology come to play in the context of agile software development an additional set of theories are presented next.

1.2.1 Agile software development: A short overview

The manifesto of agile software development already described and introduced in the Introduction Section of Paper 1 (Chapter 2) has twelve principles connected to it. These have become a popular way of defining "agility" and are as follows [5]:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, 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 to the shorter timescale.

4. Business people and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

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.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

The effectiveness of agile software development has been unproven for quite some time. However, a recent study from 2015 including 1,002 projects shows that agility does contribute to project success [6]. In a study by [7] many agile practitioners think that, in order to be truly agile, an organization needs to select practices to implement the principles in their own context. In order to understand how such practices can implement the principles we will now give a description of some of the concrete practices promoted in industry.

Agile methods (processes) Agile methodologies can be seen as an approach rather than a technique that mostly change the culture and values behind managing projects. There are some more concrete agile methods, but they all basically share the same values as described earlier. However, in order to understand how these methods work in practice, we will now shortly present some of the agile practices and how the values are implemented.

eXtreme programming (XP). eXtreme programming was the first method created by the agile community and is the most researched method [8] and considered as a relatively strict and controlled. The practices that implement the agile principles are [9]:

- The planning game. In the beginning of each iteration, the team, managers, and customers meet and write requirements in form of user stories (written in clear natural language and in a way that everybody can understand). During these meetings the whole group estimates and prioritizes the requirements.

- Small releases. Working software is up and running and delivered very fast and new versions are released continuously, from every few days to every few weeks.
- Metaphor. Customers, managers, and developers model the system after a constructed metaphor or set of metaphors.
- Simple design. Developers are asked to keep design as simple as possible.
- Tests. The development is test-driven (TDD), i.e., the test are written before the code.
- Re-factoring. The code should be revised and simplified over time.
- Pair-programming. All code is written by having two developers per machine.
- Continuous integration. The developers integrate new code into the system as often as possible. However, all code must pass the testing otherwise it is discarded.
- Collective ownership. Developers can change code wherever necessary and the overall code is assessed.
- On-site customer. A customer is in the team all the time to answer questions so the team always works according to what is needed.
- 40 hour work week. The team works with a sustainable pace defined as a 40 hour work week. The requirement selected for each iteration should never mean that the team needs to work overtime.
- Open workspace. The team should be collocated and fit in the same room. The layout of the room should make cooperation and communication easy.

Scrum. Scrum based on XP and is one of the more common methodologies and is built on embracing change and do what it takes to deliver value[1]. In Scrum the project has a prioritized backlog of requirements and use iterative development (called "sprints") to get basic working software for the customer to view as soon as possible. Scrum uses self-organizing teams that get coordinated through daily meetings called "scrums". The manager is called a "Scrum Master" to clarify that it is a facilitating role and not a directive one.

The Scrum methodology consists of three main phases: Pre-sprint planning, sprint (iteration), and post-sprint meeting. All work to be done is kept in a "release backlog" where from requirements (user stories) are taken to the current "sprint backlog". The requirements are usually broken down from a higher abstraction level when the sprint backlog is made. The actual sprint (usually 2–4 weeks) is when the implementation is performed. Here, the sprint backlog is frozen and the team "sprints" to complete what was planned. The

[1]The word Scrum is borrowed from rugby and is the situation when players from each team huddle closely together and plan what to do next in order to advance down the playing field.

team members choose tasks they want to work on themselves. "Scrum meetings" also called "Daily scrums" are 15-minute meetings every morning were the team members check status, report problems, and keep the whole team focused on a common goal. The post-meeting is done to evaluate the process and demonstrate the current system. One important aspect of Scrum is to have small working teams in order to maximize communication, minimize overhead, and maximize the sharing of informal (or tacit) knowledge. The team should also agree and be able to define when something is considered "done" [10].

Lean and Kanban. The flexible project management techniques and focus on customer value is not new. Within lean manufacturing these aspects have existed a long time (for more information about lean manufacturing see for example [11]). Many companies combine the process of Scrum with Kanban (Scrum-ban). It is important to note that Kanban is a signal card to pull products through the process within Lean production but has become a software development tool itself [12]. Scrum is a more strict process and can be modified by changing the WIP (work in progress) in each sprint into being connected to the work-flow state to prevent too much WIP. Kanban also allows adding items within each sprint. Another aspect is to change the sprint backlog owned by the team into a Kanban board with multiple teams with work-flow state instead. The Kanban board is never reset after a sprint and can be followed over time, and is also less dependent on collocation. Scrum only allows three different roles of the team, while Kanban does not have a limit. Therefore, larger teams in larger organization with a diversity of specializations often use Kanban or Scrum-ban when possible [13].

Crystal. We will not describe the Crystal methodologies in detail but, generally speaking, they are built on the assumption that the main problem in software development is poor communication. Crystal focuses on people, interaction, community, skills, talents, and communication as main effects on performance [14].

As can be clearly seen in the agile principles, these are a very high-level description of a work environment. Agile software development is an ambiguous concept with descriptions on various levels of abstraction. Many of these are obviously connected to work and organizational psychology. The problem is that these psychological aspects are not described in detail in the methods (processes). This means that this dimension is left out for practitioners to figure our for themselves to a large extent. In order to clarify the contributions of this theses we will now give a description of organizational change, development, and culture in connection to agile software development to see how we can use what is known in work and organizational psychology to filter what could possibly be useful to apply in the SE context.

1.2.2 Agile culture and agile practices

The organizational (or cultural) iceberg metaphor is highly relevant for agile management just like any other human group endeavor. To only focus on process, no matter if it is on waterfall methods or agile practices, is to only look at the peak of the iceberg. Basically, organizational culture can be divided into

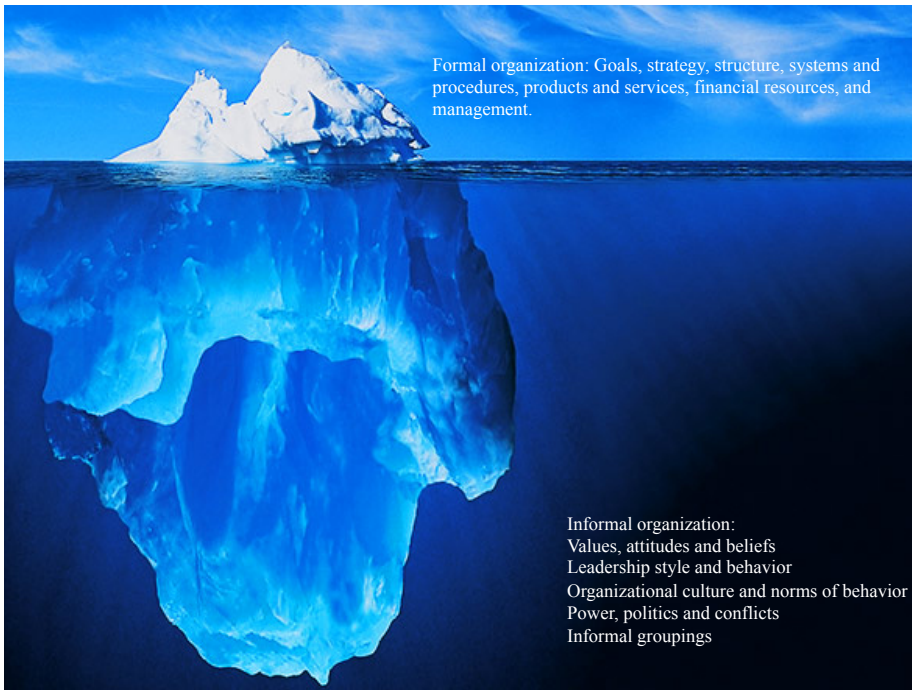

Figure 1.2: The organizational iceberg of culture. Adapted from [15].

two levels. The over-the-surface aspects are tangible and visible behaviors and artifacts in the organization, but culture is not that easy to deal with. Under the surface, there are psychological factors like expectations, values, etc. that trigger behavior. These aspects are not easy to see as an observer of the system (see Figure 1.2). The main message of the iceberg metaphor is that what we see in an organization (logotypes, greeting, communication patterns) is only a tiny part of the culture and in order to change it, we need to deep-dive into the water and understand the underlying factors of the visible behavior.

There is some research done in agile software development that point to this mistake (see e.g. [16, 17]). The agile community often separate agile principles from agile practices, as previously described.

With the lens of the iceberg metaphor it is clear that all the agile principles are a dive under the surface. This metaphor makes it clear what the differences are between agile principles (being agile) and agile practices (doing agile). This is also connected to the lack of need for agile maturity models for practitioners [18]. [19] also found it useful to sort organizational observations to further understand agile transformations in companies. Practitioners need tools for dealing with culture and not only structures for measuring it. The more formal methods focus on top-of-the-iceberg aspects whilst other models (like [20]) blend agile principles and practices in their assessment. Also, according to [7], 64.6% of 326 experienced agile practitioners stated that the reason why agile principles are valuable is:

> "/.../ all agile teams choose among software development practices,

but, if they want to be agile, they should choose practices that are in line with the principles."

If we look at the agile principles again we can see that the practices do not really specify the psychological aspect of their implementation. These aspect will be a key ingredient in the success of an agile transition. Below are the agile principles again with the reference to what psychological aspects need consideration:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. —What is customer satisfaction and does it depend on any psychological aspects? Surely (or sadly), customer satisfaction is far from just finding the best technical solution.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. —Who welcomes the change and when? In large organization the hierarchies and power structures hamper this flexibility.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. —What is needed by a company to succeed with this?

4. Business people and developers must work together daily throughout the project. —If they work together they need roles, goals and a process which demands good communication, conflict resolution etc.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. —How are trust and motivation created?

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation. —How do people communicate effectively?

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. —How can this be achieved when a project manager, for example, is held accountable for an important deadline given from top management?

9. Continuous attention to technical excellence and good design enhances agility. —How and what training is needed to achieve this, and what about group intelligence?

10. Simplicity –the art of maximizing the amount of work not done– is essential. —Do developers have the mandate to create simple solutions and simplify complex solutions if needed? Who decides what is important?

11. The best architectures, requirements, and designs emerge from self-organizing teams. —How do we build self-organizing team? Can we just tell people to self-organize?

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. —How can human behavior be changed?

This means, we need to start with a below-the-surface plan and then select practices to support those organizational changes. If we try to change the culture by changing visible processes (or in other words, using "hard" solutions for "soft and messy" problems, as they are often called in organizational change theory) we will surely fail [21]. We will now describe what this means in more detail in connection to agile software development implementation.

Hard and soft solutions of organizational change in connection to agile software development There are mainly two different aspects of any organizational change, the content of change and the context where it happens. The management ideas are often believed to be generic and we are taught to see the similarities of all types of organizations, instead of their differences. We think of a world full of organization instead of unique operative units. In order to translate organizational ideas they have to be "decontextualized" and contextualized again in a new organization [22]. A key to implement new management ideas is to have what [22] calls translator skills. One must have knowledge about the context in which one tries to implement new methods. A great problem when using generic methods is that focus always is on mean value of success. This knowledge says very little about how well a method works in one specific case [22]. This is also a reason why organizations often measure their agility in their own adapted way, i.e. the measurement models do not take the context into account enough [23].

What is a hard solution to a soft problem? In order to understand the effect of this, we must define what the differences are. The spectrum and difference are well described by [24] and can be seen in Table 1.1.

Table 1.1: The TROPICS test [24].

TROPICS factor	**"Hard" problem**	**"Soft" problem**
Timescales	Clearly defined: Short to medium term	Ill defined: Medium to long term
Resources	Clearly defined and reasonably fixed	Unclear and variable
Objectives	Objective and quantifiable	Subjective and visionary
Perceptions	Shared by those affected	Creates conflict of interest
Interest	Limited and well defined	Widespread and ill defined
Control	Within the managing group	Shared out with the group
Source	Originates internally	Originates externally

In fact, a hard problem solution is based on systems engineering and older management ideas [24] (just like traditional waterfall methods). A software engineering process change is far from that clear to the organization. Also, the cultural changes described in the agile principles and research conducted by e.g. [16, 17, 19] shows that agility is undeniably a soft issue as well as a hard one. In addition, the aspect of face-to-face communication, also stated as utterly important in the study by [7], shows that agility is only possible if people meet face-to-face. This also shows the complexity of introducing such methods and habits, core values, beliefs, priorities, politics, attitudes, perceptions, and assumptions are extremely hard to change without real life interactions.

Hard problems solution can be, for example, a simple change in an IT support system used by employees or acquisition of equipment or maintenance of supplies needed for the work place. A hard systems methodology of change usually has a description phase (with an analysis of the situation, identification of objective and constraints, and how to measure a successful change are included), options phase (evaluate different option compared to the performance measure), and finally, an implementation phase (carry out and evaluate the changes with given measures).

The solution to soft problems can be called an OD (Organizational Development) approach. An almost too accurate description of OD to agile development processes was written by [25] (first edition out in 1973). They describe the OD process as:

> "A long term effort, led and supported by top management, to improve an organization's visioning, empowerment, learning and problem-solving processes, through an ongoing, collaborative management of organization culture – with special emphasis on the culture of intact work teams and other team configurations – using the consultant-facilitator role and the theory of technology of applied behavioral science."

All the agile development lessons learned from success stories the last decade are there. The long term cultural effort, the executive buy-in, the empowered team members, the collaborative team environment, the facilitator role, and the recent findings of the usefulness of applied behavioral science [2, 26, 27].

The OD approach is considering the whole system as well as its parts. Figure 1.3 shows the assumptions for the OD process of change. We will not describe all the components in detail, but the main idea is to take a systems perspective of change and cover all different aspect of the change (from a psychological point of view as well). Therefore the change strategy should be broad and include:

1. Goals for the change in connection to the company goals, decision-making structures and how problems are solved in the organization.

2. The usage of behavioral science to change team behavior, use experimental learning techniques (an old name for learning-by-doing), and action research to implement the change (an old name for continuous improvement).

3. A planned long term and ongoing approach to the change.

4. Top management buy-in and dedicated change agents that create and communicate a climate for change.

5. A value systems that is humanistic and promotes open relationships.

6. A team approach that recognizes team processes and and the teams' interdependence.

The first step is then to diagnose the current situation with regards to organizational purpose, goals, structure, culture, prevailing leadership approaches and styles, recruitment practices, career paths and opportunities, reward structures and practices, individuals' motivation and commitment to their work and organization, employee training and development provision, intra- and intergroup relationships etc. The second step is to develop a vision for change. One does not convince people without meeting them or showing them why the change is important. The vision is the core values and the end goal of the change (the desired state). After this, there is a substantial work with gaining commitment to the vision and the need for change. This is where the hard solution approach fail with devastating consequences. Unless concerned – and to be involved in the process – are consulted and have been a part of the creation of the vision they will have little incentives for "buy-in". At this state one can not communicate too much because the more information the group members get of what is going on, the more will they back up the process. One common mistake is to focus on the people that are against you, or the ones that are not yet convinced. The key to a cultural change is to instead focus on the people that are with you and let them be bearers of the change culture. The forth step is to develop an action plan and have change agents that help the process. In the successful agile examples these change agents (that help with the change and let managers focus on day-to-day issues more) are often called "agile coaches". To have a process change facilitator is key to success in OD. The fifth step is to implement the change, assess it, and reinforce it. The latter means that the change needs to be institutionalized to be long-term [15].

We have now motivated why it makes sense to view a transition to agile software development as an organizational development (OD). We will now describe which methods where used in this thesis and how the research was conducted in order to answer the overall research question.

1.3 Method

This section will first give a more philosophical reflection on scientific discovery and then present what methods were used for each appended paper and why.

Research – The search for truth Science has from the beginning contributed enormously to the development of mankind. We have successfully observed the world and created models that help us understand and predict a diversity of events in the world, such as describing waves [28] or the photoelectric effect [29]. However, it is important to note that our predictive models

Figure 1.3: The OD Process [15].

are only models. As the famous statistician George E.P. Box (as cited in the beginning of this thesis) said:

> "Essentially, all models are wrong, but some are useful." p. 424 [30].

The problem is that in more complex systems the deterministic models are no longer useful in the same way. This is when the mathematical models can be extended with probabilities. Stochastic models that express how likely an event is to occur then makes way more sense than setting out to describe all variables deterministically (which is often not feasible) [31].

The human mind is excellent at seeing patterns in a huge number of variables [32]. Therefore, when investigating human factors, it often makes sense to collect qualitative data and let the researchers (preferably, independently of each other) systematically look for patterns in the data set (e.g. a grounded theory approach) [33]. However, as it is always good to look at a phenomenon (or construct) from different perspectives, a triangulation is always preferred. Therefore, it makes sense to collect quantitative data as well as qualitative and use statistical methods to analyze the former, i.e. using both words and numbers in the analysis [34].

Empirical software engineering (ESE) is a quite new research field compared to, for example, psychology. ESE has come a long way and made great advances. However, we believe the field is ready for having a more scientific approach to quantitative data for human factors aspects. When a research field is new it makes sense to explore, but as [35] argues in the case of software engineering already in 2009:

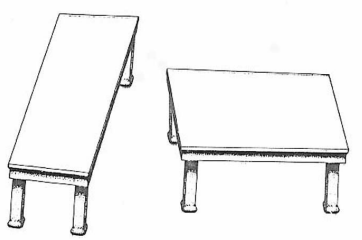

Figure 1.4: The two tables illusion. Are they of different size?

"It is time to stop 'exploring' and start experimenting." [35].

Then, we have more evidence and support for our research findings and make use of the advances in more mature fields. One challenge in software engineering is that surveys are often constructed and used without any scientific validation of the measurement. The problem is that if one skips this part and directly run statistical tests of, for example, a measurement's correlation to another, they make little sense since we do not know if we measured the right thing. A validation process is of course not only about statistical validation tests, but they should be conducted as an important step in the validation process [36].

Human perception and social science research One aspect that should be considered, when investigating human-beings in any system, is that we can never investigate the real world using people as research subjects. This cannot be done in deterministic mathematical models either, however, these models are often closer to the real world. If we use people as research subjects we will only research their perception of the context and rarely the context itself. This is an important distinction to make because people can have very different interpretations of the same situation (something witness psychology must deal with) [37]. 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 1.4 (which are of the same exact size).

However, sometimes we automatically assume different contexts, like the dress shown in Figure 1.5). 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 perception 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 and Keating in 1972 [38]). When investigating human factors in software engineering, we should always state that we describe the perception of the construct and not the construct itself, which is actually often what we are interested in anyways. However, everything is not up for interpretation and many big scientific discoveries are the truth for our world, like the evolution [39] and heuristics in behavioral economics [40], simply because we have gathered massive empirical evidence.

Figure 1.5: The infamous dress picture. White and gold or black and blue?

Table 1.2: A correlation matrix.

	A	B	C	D	E	F
A (reading)	1.00	.60	.50	.15	.20	.10
B (vocabulary)		1.00	.50	.15	.10	.10
C (spelling)			1.00	.10	.20	.15
D (addition)				1.00	.60	.60
E (subtraction)					1.00	.60
F (multiplication)						1.00

If we want to investigate these types of issues we can look at other fields that have been dealing with social science for more than half a century. If we use humans and their opinions in research we only investigate their perception of what is happening in the organization, as previously stated. Even if this is the case, we still need to check that items used to measure a construct manage to circle it somehow, i.e. items that are different but still correlated in relation to the construct under investigation. In order to do this we need two things; first, a reasonably large sample representative of the population (and large enough to remove individual and cohort bias), and second, make sure our items are correlated and pinpoint our construct of interest.

To simplify this explanation, let us look at a simple example. If a test (e.g. a survey) gives the correlation matrix in Table 1.2 the corresponding factor loadings would be the ones showed in Table 1.3.

How to obtain the factors in a factor analysis is an advanced mathematical method and we will not go into details on how the calculations are conducted. However, the main reasoning behind the technique is that if p is the number of

Table 1.3: The corresponding factor matrix with factor loadings.

	Factor 1	Factor 2
A (reading)	.70	.15
B (vocabulary)	.60	.15
C (spelling)	.60	.10
D (addition)	.10	.70
E (subtraction)	.10	.65
F (multiplication)	.05	.60

variables $(X_1, X_2, ..., X_p)$ and m is the number of underlying factors (the model presumes we have underlying factors) $F_1, F_2, ..., F_m$ is that specific variable's representation in the latent factors. Each measured (or observed) variable is then a linear combination of the factors and reproduce maximum correlations: $X_j = a_{j1}F_1 + a_{j2}F_2 +, ..., +a_{jm}F_m + e_j$ where $j = 1, 2, ..., p$ and a_{j1} is the factor loading of the j^{th} variable on the first factor and so on. The factor loadings can be seen as weights (or contrast) of a linear regression model and tell us how much the variable has contributed to the factor. There are different extraction techniques for factor analysis and if the error variance is included it is called a Principle Component Analysis since we use all variance to find factors. If the error variance is excluded it is often called a Principle Axis Factor extraction, but the output pattern matrices are very similar [41].

In this case we probably had reason to believe that there were two factors (or constructs) based on the variables in the study. If, for example addition and subtraction were uncorrelated in our result, we would have reason to doubt our measurement of the construct of mathematical skills. Even if a construct like "mathematical skill" is ambiguous, we still need to make sure our subset (like mathematical operators –addition, subtraction, and multiplication–) are valid. In this simple example we would have empirical support that items A, B, and C describe one construct and D, E, and F another, which also makes sense (i.e. high face validity).

We can extend the same reasoning to that of "agility". Even if we do not have a clear definition of this term we can still research agile practices or behavior. For example, if we want to research Integration Testing or Retrospectives, we must use items that are different but correlate in a satisfactory way.

The whole discussion of "if we measure what we think we measure" is dealt with in most papers in the Validity Threats section, which we also do in the appended papers to this thesis. But what is validity? When it comes to tests in human systems we cannot just look at the measurement tool itself but also the context and interpretation of test scores, like in the definition of validity by [42]:

> "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 means we always validate the usage of a test, and never the test itself.

When soft issues are investigated in psychology they are, of course, often analyzed using both quantitative and qualitative data. However, after a more exploratory investigation (usually through qualitative case studies) we need to proceed and collect empirical evidence of the phenomenon (or construct) we found and see if numbers support our ideas. Many researchers within software engineering also have a misconception of external validity and practical usefulness [43]. Of course we need a holistic view of validity and studies using quantitative data sometimes get undeserved credibility since the mathematical methods alone can seem advanced and serious. However, this is also what we

see as a danger in software engineering survey research. If we create a survey and skip the statistical validation procedures we do not know what we measure. The statistical validation is only one aspect of the validation needed, but we should at least make sure that part supports our hypotheses. Otherwise it would be like decorating our ship with fancy canons without having checked if it even floats, which has happened before [44].

Getting "enough" data is always a tricky aspect of statistical tests. When it comes to factor analysis (see Table 1.3), the sample size needed is dependent on e.g. communalities between, and over-determination of, factors. Communality (or Internal Consistency) is the joint variables' possibility to explain variance in a factor, which can be calculated by, e.g., the Cronbach's α [45]. Over-determination of factors is how many factors are included in each variable [46].

Methods used in this thesis The statistical method of factor analysis presented in the previous section is one way of checking if the numbers support the idea that a test really measures what we hope it does. This will not replace other aspects of validity but we do not see any disadvantages with collecting empirical evidence for surveys used in research. In the field of psychology, researchers need to be very careful stating that surveys, that have not been scientifically validated, give any kind of 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 2 (Chapter 3) uses such a statistical test (i.e. a factor analysis) on an agile maturity model.

Of course getting 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. Such an exploratory study was conducted regarding work motivational aspects in the interface between agile teams and others in a more traditional surrounding organization in Paper 1 (Chapter 2). However, this implies that we cannot know if the findings are true for any other organization, but is nonetheless interesting as a new angle of work-life and research in the field of agile software development.

Doing the other way around (i.e. using only quantitative data and not qualitative) when deeper understanding would be useful, is not preferred either. In Paper 3 (Chapter 4) we only collected quantitative data in form of a survey and ran a correlation analysis, which shows an initial understanding of that something probably needs more investigation. That is why we conducted, and added qualitative data (in addition to more quantitative data as well), to the research presented in Paper 4 (Chapter 5).

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: Work motivational challenges regarding the interface between agile teams and a non-agile surrounding organization

The process of agile software development implicates a larger focus on teams and communication, which means that psychological aspects (of e.g. group psychology) play a key roll in the effectiveness of teams and therefore the quality of their output. Paper 1 presents a case study on work motivational aspects of introducing agile teams in a more traditional organizational structure. This increases the understanding of what happens to group members' work motivation and helps researchers and practitioners to be aware of what happens on a psychological level to people assigned to these teams. The results show that there are work motivational aspects to consider. One aspect was team members not feeling appreciated when delivering their work, due to synchronized feedback loops with the surrounding organization. When the team delivered something great and wanted to celebrate, everything was quiet. When a milestone was reached in the overall project plan, they got positive feedback on deliveries already weeks old. Team members also expressed frustration when working on "normal" teams after, or during, the agile project since the pace was slower and the environment much less responsive. Therefore, companies need to adapt their feedback loops to the new agile teams (or the other way around) so they are synchronized. If not, team members will probably feel unappreciated due to this lack of important feedback. The second useful aspect for practitioners is to prepare their employees of the difference between more agile teams and their more traditional settings. This might prevent them from, or at least be prepared for, frustration when pending between different team set-ups.

The main contributions of Paper 1 are:

1. A new perspective of teams' motivation when transitioning to agile software development processes in larger companies.

2. How to prevent these negative motivational aspects.

This paper helps us to answer RQ1: "How can we use work motivational psychology to increase our understanding of agile teams in a non-agile surrounding organization?" There are many aspects of work motivation and some have already been studied (see [2] for an overview), however, we conclude that there are aspects of work motivation still unexplored and Paper 1 contributes with adding a perspective on the dynamics of agile teams in larger organizations.

1.4.2 Chapter 3/Paper 2: The prospects of a quantitative measurement of agility: A validation study on an agile maturity model.

In order to further investigate how agile practices and their adoption are related to work and organizational psychology we need to measure agility somehow. The issue of how to measure agility with high validity is not thoroughly researched in software engineering. Therefore, as a first step of assessing agile

maturity measurement, Paper 2 presents a validation study on such a measurement tool (including a pretest case study with a team). The method used for evaluating the tool is taken from how psychology scales/measurements are developed in the psychology field with focus on rigorous and well-used statistical tests. The results show that the tool under validation needs more work and we discuss the difficulty of measuring agility in such a way.

The main contributions of Paper 2, as stated in the highlights for 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 result).

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

5. Tradeoff between quick quantitative versus time-consuming contextual assessments.

This paper helps us to answer RQ2: "How can we use quantitative statistical tests (used in psychology research) to validate surveys of agile maturity?" Some publications make use of these methods (see [6, 47] for an overview), however, we conclude that these methods are useful for larger survey research in software engineering generally. In the quest to find measurements for agility, such larger studies with tests like factor analysis is a must in order to move forward.

1.4.3 Chapter 4/Paper 3: Group maturity and agility, are they connected? – A survey study

As a first step to investigate the connection between agility and group development (i.e. check if an agility measurement has significant convergent validity with the group development questionnaire), Paper 3 correlates overall means of a poorly validated agility measurement to a thoroughly validated group development measurement taken from social psychology. This perspective of agility helps define the ambiguous concept of agility somewhat since a mature team in social psychology and agile teams go hand-in-hand. It is hard to draw conclusions of causality from only a correlation analysis. Maybe group maturity is a requisite for agility or that agility drives or enables group development. Or we have some other factors influencing both. A fourth option could be that the found correlation is a pure coincidence. However, we strongly believe that all the success stories on agile teams includes entirely overlapping descriptions of a mature team. Therefore, we think it helps both researchers and practitioners to view agile teams as mature teams. Then many of the agile practices can be seen as enablers of group development, which leads to higher understanding of what happens in teams as well as increased predictability. In

order to investigate these connections and causalities we collected more data and conducted additional interviews included in Paper 4 (Chapter 5). That manuscript is an extended version of Paper 3 in Chapter 4.

The main contributions of Paper 3 are:

1. Some agile methods could be seen as enablers of group development.
2. We defined an "agile team" as a Stage 4 group, which gives us tools to create them (Stage 4 is correlated to other effectiveness measures in other fields).
3. New groups cannot be agile (e.g. continuous improvement cumbersome), if agility is defined in such a way.
4. Teams will adopt agile practices differently depending on their group stage.

This paper together with Paper 4 helps us to answer RQ3: "How can we apply group developmental theory to increase our understanding of agile teams?" This is the most practical of all the contributions of this thesis. We are the first researchers who showed empirical evidence of connections between agile teams and group development. These studies show how software engineering could use knowledge from social sciences instead of inventing the wheel allover again. Groups have been researched in psychology for almost a century and to focus on where SE is different instead seems like a better idea. What the agile software development body of knowledge was lacking were guidance on how to create these mature teams, from a group psychology and developmental perspective.

1.4.4 Chapter 5/Paper 4: Connections Between Group Maturity And Agility: A Quantitative and Qualitative Investigation at Eight Companies

Paper 4 is an extended version of Paper 3 where we added 32% additional quantitative survey data and ten interviews. The reason for this was to investigate the causal relationships further and get more depth into 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 psychology of building agile teams. This means that teams will adopt 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.

Paper 3 includes only an overall correlation and this publication adds the following:

1. 32% more quantitative data.
2. A different division of agile factors as presented in Paper 2.
3. Ten semi-structures in-depth interviews with agile overall responsible coaches/managers from seven multinational companies.

4. An integrated analysis and discussion of both the new quantitative analysis and the qualitative interviews.

5. Suggestions of how to work with these psychological aspects in agile software development.

The main contributions of Paper 4, as stated in the highlights for the publication, are:

1. Found correlations between measurements of agility and group maturity.

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

3. Helps define agility as a mature group as described in group psychology.

4. Implications for how agile teams can work with group development.

In the next section we will discuss the presented papers and see how they fulfill the overall research objective.

1.5 Discussion

The presented papers show that there are clearly ways that agile software development can use work and organizational psychology to increase the understanding and predictability of the developmental processes used. Not only directly applicable knowledge from extensive research but also what scientific methods the more mature field of psychology uses in research advances. This interdisciplinary field has been proposed to get the name "behavioral software engineering" by [2]. The authors also argue for the importance of using the systems perspective we described is a key ingredient in organizational change research [15].

The field of work and organizational psychology is huge and some research has been done on these aspects within software engineering (see [2] for an extensive overview of what has been done). However, there are many gaps and this thesis fills some of these in the agile domain. In order to sort where this research fits in, in such a broad interdisciplinary field, we will discuss the findings and contributions using the organizational development (OD) map presented earlier. This means we will fit our research result into the context of an organizational change to also see where in this "systems approach to change" more contributions can be made. Figure 1.6 is that map with added papers from this thesis and intended future research. Many aspects of that map has research conducted, even in the boxes with the papers from this thesis. We want to use the map to place the papers in a context and we find it useful to look at an agile transition as an organizational change as described in management since the 1970s, i.e. a sub-area of what [2] calls behavioral software engineering (BSE). As mentioned in the introduction we use the term "organizational development" as a subcategory of "work and organizational psychology" that focuses on organizational changes. We also use the agile software development context but the applicability of behavioral software engineering exceeds this category and is applicable to many more aspects of software engineering.

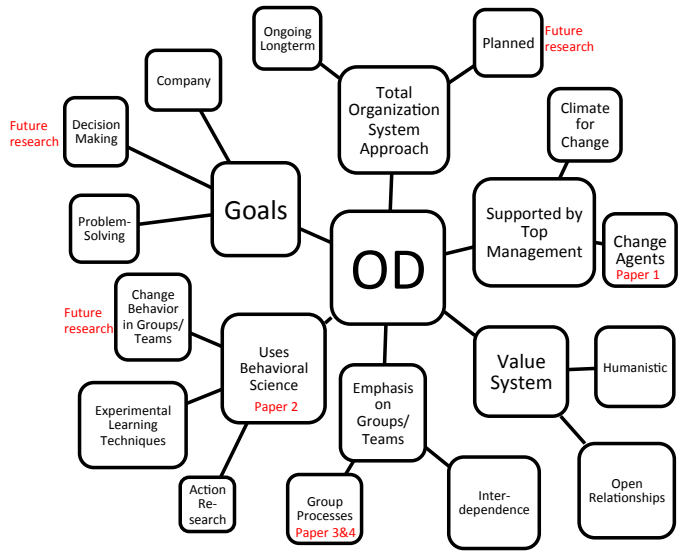

Figure 1.6: The OD process in connection to our research on transitioning to agile software engineering.

Paper 1 is about work motivational aspects when group members are pending between agile and more traditional project settings. Therefore, these group members become change agents for the agile organizational change at the company. In a transition to agile, as seen from an OD perspective, such change agents are essential for a change in culture.

The aspect of using behavioral science for change means that we need to use evidence-based approaches to change the behavior when we aim at changing the culture. In order to know what to do in connection to the wanted "agile" behavior, we need validated measurement tools. Paper 2 is an important step in obtaining such a tool.

Papers 3 and 4 obviously fit into the group processes box. It has been known for a long time that groups are essential when changing a culture, and not the individual. In many technical fields that include human factors, researchers often conduct personality tests due to their straightforwardness and quantitative output. Such tests give numbers possible to correlate to other measurement, which might be interesting from the research perspective of describing what personalities are most present in what professions etc. However, this research will not help in changing culture or putting together teams since teams have a collective intelligence [48], and looking at personalities is simply the wrong abstraction level. The research conducted during 40 years on personality tests in software engineering also give different result in a meta-study from this year (2015) [49], which we believe supports this statement. In addition, practitioners think the group and organizational levels of human factors are more important than the individual aspect [50].

In order to guide work groups (teams) through their agile adoption, we must look at group-level issues, norms, culture etc. Papers 3 and 4 is a first step that connects group maturity to agility, which is a prerequisite for creating

guidelines for teams on different maturity levels in connection to their selected agile practices and behavior.

Being aware of all the mentioned aspects of an agile implementation will help guide practitioners and researchers in implementing or researching these concepts further. The results of these articles, and specifically the result that agility and group maturity are connected, are not surprising on the one hand. On the other hand, only anecdotal and success stories have described the different needs to different types of teams (like [14] for example). The difference between those descriptions and the contribution of this research is the empirically gathered evidence for how different types of teams (i.e. teams on different group development maturity levels) will adopt agility differently. To highlight the importance and relevance of looking at psychological aspects in the software engineering domain will hopefully convince both researchers and practitioners of the usefulness of adding such a dimension.

The overall research objective is therefore fulfilled because we have shown that methods and models from work and organizational psychology are indeed applicable, especially to the context of agile software development. Methodological techniques (like the statistical validation of Paper 2), aspects of how work motivation plays a part in larger organizations who introduce agile teams (Paper 1), and the direct applicability of group development theory when building agile teams, are all contributing to the overall research objective whether can we use work and organizational psychology to increase our understanding of agile software development.

1.5.1 Validity threats

Instead of listing categories of validity threats, this section will describe the actual threat we see to the different papers.

First of all, we would like to state that even the largest 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 large enough for a meta-study to have such claims of external validity. See for example [51], where they used 225 studies to conclude that active learning outperforms classical lecturing with regards to student performance. This article provided incentives for us to apply a new pedagogical strategy in the Empirical software engineering course at Chalmers and University of Gothenburg. The point is that new concepts in exploratory research (like that of Paper 1) is 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 case studies since a validation of the possible causal relationships is included in the design. Paper 2 (Chapter 3) shows one aspect of testing the validity of that maturity tool, but could also not be valid in itself of course. However, it shows clearly the tradeoff between such a quantitative method and other qualitative contextual assessments that might be relevant for consultancy, but does not contribute as much anymore to science. Paper 3 (Chapter 4) does not show high external validity in itself since only two companies participated and the data collected was only quantitative. However, the results showed a correlation and Paper 4 (Chapter 5) gives a deeper

analysis and an increased possibility to understand the behavior. In order to state if group maturity is a prerequisite for agility more and larger studies are needed, preferably longitudinal studies to draw conclusions about causality. The construct validity of "agility" is probably the largest threat to the papers in this thesis. However, this thesis contributes with defining the subcategory of "agile teams" as mature (Stage 4) teams as described in social psychology in order to, at least, pinpoint what agility means on a team level. The correlation shown in Paper 3 (Chapter 4) could be seen to strengthen that agile maturity tool since it had significant criterion validity to another validated tool. If agility means at least some aspects of group maturity, the maturity model captured some aspects of it, since the measurement is correlated to the Group Development Questionnaire (GDQ). Also, as stated in Paper 3 (Chapter 4), we could possibly also have internal validity threats in 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 quantitative approaches are usually considered lower than collecting quantitative data. The standard validation method used in Paper 2 can easily be repeated on another data set. When it comes to the interviews etc. that we conducted, the perception of the researcher introduced social bias (as described in the method section). This means that these studies are less reliable since they are difficult to replicate. However, the combination of a quantitative survey and qualitative interviews triangulates the issue of group development and agility in Paper 4, which then mitigates and increases the validity overall.

All in all, the fulfillment of the research objective of finding areas where models and methods from work and organizational psychology can be used to increase our understanding of agile software development has high validity due to a number of reasons. First, we used a qualitative approach where exploration was needed (Paper 1), we used a thoroughly validated measurement of group development (Paper 3 and 4) and we applied well-used statistical test in our validation study of the agile maturity model (Paper 2). This provides evidence of the usefulness of methods and models taken from the field of work and organizational psychology.

1.6 Conclusions and future work

This paper set out to investigate if methods and models from the field of work and organizational psychology can be used to increase our understanding of agile software development. Through applying both some psychology frameworks (e.g. a group development model) and research methods (e.g. a factor analysis) to the context of agile software development teams, we have found that more research on psychological aspects of software engineering in general would help the understanding and predictability of what the field is dealing with. These findings are important contributions to researchers in the field of software engineering since we get a deeper understanding of what happens in these large systems of interacting people that software development organizations also are. While we have specifically focused on applicability to agile software development from an organizational development perspective, the na-

ture of the usage implies that our findings are likely to be of importance to more areas in software engineering that deals with human factors. We believe this new perspective could change the field of software engineering in the same way as behavioral economics emerged as a sub-field of economics [52]. Possibilities for future work is shown in Figure 1.6. We see potential in conducting research within strategic agile decision-making, further researching the aspect of group maturity over time in connection to agility with larger sample sizes, but also investigating individual cognitive biases in the requirements engineering context and so on and so forth.