Barriers Against Better Team Performance in Agile Software Projects

Master of Science Thesis in the Master's Programme International Project Management

YAVUZ KOZAK

Department of Civil and Environmental Engineering
Division of Construction Management
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2013
Master's Thesis 2013:122

REPORT NO. 2013:122

Barriers Against Better Team Performance in Agile Software Projects

A Qualitative Study To Identify Factors That Can Decrease Performance of Agile Software Teams

Master of Science Thesis in the Master's Programme International Project Management

Yavuz Kozak

Department of Civil and Environmental Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2013

Barriers Against Better Team Performance in Agile Software Projects
A Qualitative Study To Identify Factors That Can Decrease Performance of Agile Software Teams
YAVUZ KOZAK

Technical report no 2013:122
Department of Civil and Environmental Engineering
Division of Construction Management
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Civil and Environmental Engineering Göteborg, Sweden 2013

Barriers Against Better Team Performance in Agile Software Projects
A Qualitative Study To Identify Factors That Can Decrease Performance of Agile Software Teams

YAVUZ KOZAK
Department of Civil and Environmental Engineering
Division of Construction Management
Chalmers University of Technology

Summary

Agile methodologies evolved to cope with the changing requirements in management of software projects, in an effort to increase bring business value of products and create higher client satisfaction. At the heart of agile methodologies lies the agile project teams. Success of agile projects highly depend on the performance of project teams. In contrary to the driving forces that foster performance of teams, there are barriers that limit their performance and productivity.

In this master thesis, factors that limit the performance of teams operating in agile software projects are studied. A literature review approaching the research question from three dimensions (software projects, agile methodologies and teamwork) is made. Qualitative data, collected from interviews with 8 agile project managers is used to identify and present various barriers grouped under three main topics: (1)Interaction and personal perspective, (2)material perspective and (3)agile process perspective.

It is concluded that performance of teams in agile projects depend on many factors and there are different barriers that prevent teams from progressing more efficiently throughout the project. Barriers identified in this thesis are interpretations of the participants of interviews and generalisations may not apply to all teams in the industry.

The report is written in English.

Keywords: Agile, Software Project Management, Team Performance, Barriers

Preface

This master thesis identifies barriers that decrease performance of teams that operate in agile software projects. Thesis was carried out between January and September 2013, at Chalmers University of Technology, Construction Management Department. Thesis was supervised by Inger Bergman from Chalmers University of Technology and Dr. Claudio Benghi from Northumbria University.

Author of this thesis would like to thank his supervisors for their guidance, family and girlfriend for their support throughout the thesis work. He would also like to thank all the project managers that he has interviewed for their valuable contributions on the context of the thesis.

Yavuz Kozak,

Sweden, September 2013

Table of Contents

Table of Figures

1 Introduction

Software industry has grown significantly in the last decade. BSA (no date (Accessed: 12 April 2013)) report on software industry reveals that "software and related services sector experienced a real annual growth rate of 14%" in U.S.(which corresponds to 46% of the world software market(BSA, no date (Accessed: 12 April 2013)) between 2006 and 2007. Together with the growth in the industry, the demand for faster and more precise software solutions increased. This growth of demand and industry brought the necessity to change the way software projects are managed. Shorter deadlines, more complex requirements and ever-changing client needs made it hard for traditional development approaches to cope with. (Takeuchi & Nonaka) argued in (1986) that sequential approach to developing new products are no longer able to get the job done.

Emergence of agile methods in software development was a consequence of effort to cope with the dynamic environment of software management process. Observing that "while hardware speed and network capacity have made impressive strides ... software development has not improved under the same order of magnitude"(Chiang & Mookerjee, 2004), the idea that 'software development processes should be improved' was born. Agile Manifesto was created in (2001) as a guideline for improving the software development process, giving more emphasis on interactions and flexibility. The four values of the manifesto are:

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

(http://agilemanifesto.org/, 2001)

Today, many software development methodologies that are characterized as agile are being used (Scrum, Extreme Programming, Lean etc.). These comparably fresh methodologies offer more flexibility, creativity, productivity and stakeholder satisfaction when compared to traditional methods. Zwicker, 2007, cited in Fernandez and Fernandez (2008) agrees on this argument stating that majority of the companies using agile methodologies that he polled saw more than 10 per cent improvement. However, for these methods to work as intended, extensive efforts are in several areas. Team dynamics is one of those areas since success of agile methods are highly dependent on interactions and team effectiveness. Moe, Dingsøyr, and Røyrvik (2009) state that team performance has direct relationship with effectiveness of teamwork coordination. However, they add that usage of teams does not guarantee success for the organisation. As they can be the architects of success, they can also be the source of failure. "Most software development projects fail because of failures with the team running them"(DeMarco and Lister, 1999, cited in (Acuña et al., 2009)).

Teams can be thought as systems which are influenced by positive forces such as high budget, experienced specialists in the team, high-tech tools, capable consultants, *etc*. and negative forces, which are the barriers that this research focuses on. To increase speed(efficiency) of a system(team), positive forces should be supported and negative forces should be eliminated. Sanjiv Augustine et al. (2005) stated that throughout the project, the project manager identifies problems and removes obstacles to implementation of the practices.

Inspired by the above ideas, in this study, the barriers that affect the team performance in agile software projects negatively were attempted to be clarified. Focusing on two major agile methodologies (XP and Scrum), obstacles that teams encounter are researched, analysed and listed under logical groups.

Following introduction chapter, literature review examines the scholarly knowledge created so far about the topics of interest. Next, methodology chapter explains the literature review and data collection methods as well as research type, ethical considerations and limitations. Results and data analysis chapter starts with a brief analysis of the data sources and continues with listing of data collected. Discussion part critically analyses the collected data under the light of literature review. Finally, conclusion chapter makes an overall summary of the research and suggests further research.

2 Literature Review

This chapter focuses on the previous works of scholars about subjects related to the research question. The research question was approached from three dimensions:

1. Software project management,
2. Agile methodologies,
3. Teamwork.

For this aim, software project management, agile project management and teamwork will be discussed briefly in the following sections. Next, explicitly stated and possible barriers found in analysed literatures will be analysed.

Information gathered from academic journals and textbooks were organised in a logical manner to give the reader an understanding of why the research question is of importance, which angles of project management should be considered and findings in previous research that are related to the research topic.

2.1 Software Project Management

Association for Project Management (2006) defines projects as unique, transient endeavours undertaken to achieve a desired outcome. Ahmed (2012) defines project management as starting an activity to achieve some stated goals using limited resources, budget, and time. Software projects are born with specific or unspecific requirements from the client, which in time evolves to a software product (product was depicted as source code in (Ahmed, 2012)). These projects are run by individuals or teams-which can be multi-disciplinary-that try to implement all the specified goals with their human and material resources in a given time and with the least amount of defect. The end product is expected to satisfy the client in terms of functionality, usability, performance and delivery time.

2.1.1 Key Concepts in Software Projects

Client Requirements

Today, most of the organisations require software. From simple web-pages to online sales pages of retailers, or from customer databases to enterprise resource planning systems organisations require software of different complexities for different needs. Although software industry provides standardised solutions to the market, most clients require specialized software that complies with their organisational structure and addresses their specific needs.

It is rarely the same what client defines in the beginning and what he expects at the end of the project. In the beginning of the projects, requirements being not complete or clear the client has a set of needs and a vision of the final product. These elements are usually changed by the client because of several reasons such as

- Client may not need some of the functionalities any more,
- Client may want to put extra functionality,
- Some functions client described may not address to its needs as client expected,
- Client may want change in design,
- Requirements may be perceived differently by the project team.

Therefore, as the project progresses, existing requirements may change or new requirements and constraints may be added to the projects.

One of the major goals of project teams is to satisfy the client by implementing the requirements set-and in time may be changed-by him. Therefore, requirements and client are two of the important factors in software projects.

> *"For the truth is, the clients do not know what they want."*
>
> *(Brooks, 1995)*

Project Team

One of the limited (and the most important one) resources of projects is the project team. Performance of teams (thus projects) largely depend on the cumulative skillset and cohesion of the team members. Deeprose (2001) argues that teams are differentiated from other work groups such that team's members have a common purpose and they need each other to attain it. Moreover, they share and divide roles and responsibilities to accomplish the agreed on tasks.

Software projects require combination of different skills in the team. Developers, architects, testers and managers are common for most of the software project teams. Performance of projects depends on the level of communication and cohesion between the team members rather than their number. Brooks (1995), argues that effort and progress are not interchangable, therefore performance of teams are not linearly related to the number of team members. In addition, he claims that each new member added to the team brings exponentially increasing amount of need for total communication.

> *"Projects, after all, are all about people."*
>
> *(Ahmed, 2012)*

Tools

It is in the virtual nature of software programming to get involved with specific tools since the end product is a list of commands interpreted by a hardware, rather than something physical. Software projects require tools like servers, computers, development and office tools, project management tools to "increase productivity and quality, shorten delivery time and accomplish manually impossible tasks"(Ahmed, 2012).

Selection of appropriate tools and training team members on the usage of tools can increase success potential of projects.

Process

From inception to hand-over, software projects are managed using predefined guidelines and standards. These are formed by experience and research of many years by contribution from academia and industry. There are various project management processes on the market that helps project managers to plan, execute, evaluate, measure and document the projects.

Project processes include all the activities starting from project initiation to closure. Depending on the size and complexity of the projects, length and number of activities vary. All of the activities in a project are interrelated and are executed in a logical order with respect to the progression of the project.

> *“One size[process type] does not fit all [projects]!”*
>
> *(Ahmed, 2012)*

2.1.2 Agile Software Development

This research’s focus is on agile software development methods. Because of the limitations in data sources for interview, two iterative and incremental methodologies, namely XP and Scrum will be the main two methods that will be focused on. This section will make a brief explanation of the agile methods.

What is Agile Software Development

The term *agile* comes from agility, which is used as a mean to express the adaptable nature of agile software development methods. “In the centre of increased globalization is the need for project managers to have flexibility in a project system in order to be able to adjust constantly to emerging challenges and opportunities”(Fernandez & Fernandez, 2008).

Traditional (or so called sequential) methods, which are older than agile methods were being designed as series of pre-planned actions that follow one another. Sequential methods assume and require that a phase should be finished before another can start. For instance, to begin the testing phase of a project, the whole product was expected to be built and integrated. This approach has several flows related to adaptability of the process and quality of the product being closer to customer expectations:

- Planning of complex and long projects are hard and actual progress diverges from the planned progress in time,
- In software projects, customer expectations and specifications about the product change over time. Introducing those changes to the project decreases the plan’s applicability,
- Integration of the code is done at the end of programming phase in sequential methods. This leaves the project team less time to make fixes

and changes over the product to be tested. Moreover, finding and fixing bugs in the integrated code takes more time.

Agile software development methods, greatly reduced these drawbacks of sequential methods by making the development process more adaptable to changes.

Agile manifesto, which summarizes the principles of agile development was introduced in 2001. Four principles of agile manifesto are stated below.

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

(http://agilemanifesto.org/, 2001)

Following these principles in their processes, several software development methods were introduced in time. Abrahamsson et al., 2003, cited in Sanjiv Augustine *et al.* (2005) state that agile development methodologies have sought to focus on rapid iterative delivery, flexibility and working code. This research focuses on two popular methods that are used widely in software management industry. These methods are XP and Scrum, which are based on agile values and are similar in many ways. Principles and practices of those methodologies will be summarized below. Principles behind agile manifesto (see Appendix C – Agile Manifesto and 12 Principles) that forms a base for principles of the two methodologies will be included as quotations.

Principles

1. Communication

> *"The most efficient and effective method of conveying information to and within a development team is face-to-face conversation."*
>
> (*Principles Behind the Agile Manifesto*)

Like other agile methods, XP and Scrum makes heavy use of face-to-face communication throughout the project. Basic idea behind this principle is that face-to-face communication allows flow of knowledge faster and more freely.

Koch (2004) states that agile methods all prefer face-to-face communication over written means. Meetings, working environment, programming methods are designed to foster the communication between the team members. de Sousa and Almeida (2011) stated that face-to-face communication creates a greater potential for effective understanding.

2. Self-organising teams and motivated individuals

> *"The best architectures, requirements, and designs emerge from self-organizing teams."*
>
> *"Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."*
>
> *(Principles Behind the Agile Manifesto)*

Avoiding micromanagement and empowering the agile team to make its own decision is one of the key principles in agile methods. It is claimed that when teams are given more responsibility about the control of project progress, they will "organize their work in a way which suits them best to accomplish their mission"(Stober & Hansmann, 2010). Under this context, Takeuchi and Nonaka (1986) argues that agile teams operate like a start-up company. In addition, self-organising behaviour in teams is effective when individual members are motivated and knowledgeable enough to fulfil their parts in the team, without the need to take orders from someone else.

3. Early and frequent delivery of working software

> *"Our highest priority is to satisfy the customer through early and continuous delivery of valuable software."*
>
> *"Working software is the primary measure of progress."*
>
> *"Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale."*
>
> *(Principles Behind the Agile Manifesto)*

A key difference between sequential methods and agile methods is the release of a working software several times throughout the project. Improving and adjusting in every next release, the product converges to what the customer really wants. During demos, customer is asked to examine the prototype and give feedback so that they can be applied as soon as possible. Moreover, this gives the customer to add or remove features to/from the feature list in the middle of the project. Since aim of the team is towards creating a working software that fulfils the needs of customer, the end product is more likely to progress towards customer satisfaction. Fernandez and Fernandez (2008), in their research, emphasize that structure is most effective when oriented on "product" instead of "process".

4. Technical excellence, good design and simplicity

> *"Simplicity--the art of maximizing the amount of work not done--is essential."*
>
> *"Continuous attention to technical excellence and good design enhances agility."*
>
> (*Principles Behind the Agile Manifesto*)

Attitude in agile projects is towards sparing enough time for a piece of work to produce it with high quality. This decreases the need for rework, retesting and rewriting. Moreover, "when changes are required (for example, incorporating a new requirement), well structured, cleanly implemented programs will be easier to change."(Koch, 2004). Aiming for a good quality code also enhances the programmers capabilities.

In addition, keeping the code simple helps the team to avoid tasks that add no value to the project and prioritize ones that are more important. Simple design also helps programmers to test and update the code faster. "Agile project management approaches also emphasize a generative approach, where only what is needed(processes, tools, procedures, documentation, etc.) is required to be used in the project" (Fernandez & Fernandez, 2008).

5. Stakeholder collaboration

> *“Business people and developers must work together daily throughout the project.”*
>
> *(Principles Behind the Agile Manifesto)*

Agile methods rely on continuous collaboration between every stakeholder in the project. People from both the technical and business aspects of the project share their ideas and express their opinions. A difference from usual stakeholder management in traditional project management methods is that communication is done more frequently, and customer is actively included in the projects.

6. Sustainable pace

> *“Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.”*
>
> *(Principles Behind the Agile Manifesto)*

Traditional project management methods usually work with a bell-shaped pace where the work done(code written) is less than average at the beginning and end of projects (see Figure 1). Koch (2004) states that to make regular progress apparent, and avoid rush towards the deadline, a consistent rhythm should be established. Moreover, working extra towards the deadline has a chance to cause exhaustion for the next iteration. Therefore, agile teams work with a sustainable pace, trying to avoid working for extra hours.

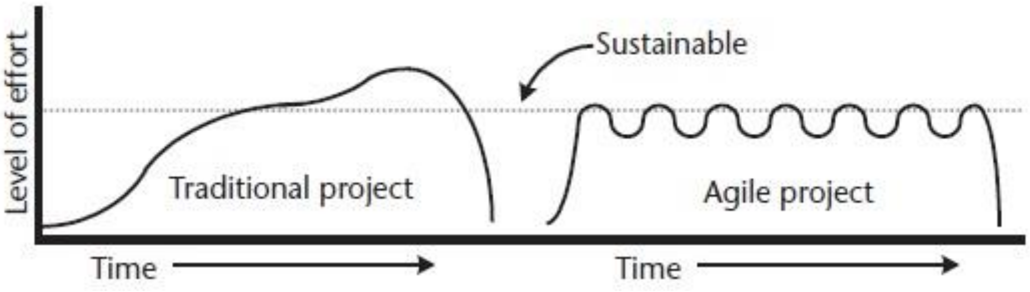


Figure 1 – Level of effort in traditional and agile projects.

(taken from (Koch, 2004))

7. Self-reflection and adjustment

> *“At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.”*
>
> *(Principles Behind the Agile Manifesto)*

Retrospectives are important part of agile methods. Evaluating past performance of self and the team and trying to adjust to perform better is essential to adapt to requirements of the project. Retrospective meetings and feedbacks from other team members serve the purpose of self-reflection in the team and adjusting accordingly. Instead of having a lessons-learned meeting at the end of the project in traditional projects, agile projects spread this throughout the project to apply needed changes right away.

8. Welcoming change requirements

> *“Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.”*
>
> *(Principles Behind the Agile Manifesto)*

Ambiguity of requirements is considerably higher in software projects. Therefore, requirements tend to change during the project. Agile projects welcome these changes since adjustment of project plan is easier with shorter and more frequent iteration plans. Even towards the end of the project, new requirements are accepted and implemented in an effort to increase the business value of the product towards client’s needs.

Practices

1. Working Environment

The environment that teams are working in directly affects their interaction level. XP and Scrum teams are sometimes situated in a separate office, sitting in close proximity to each other and having tools like meeting rooms, whiteboards to foster information flow (see Figure 2). These workspaces are “designed to maximize face-to-face conversation within the team. It also facilitates “accidental communication,” as people overhear what other pairs are discussing.”(Koch, 2004). de Sousa and Almeida (2011) also mentioned that team members believe that frequency of interaction between team members are directly related to the proximity between physical spaces each member occupies in the laboratory.

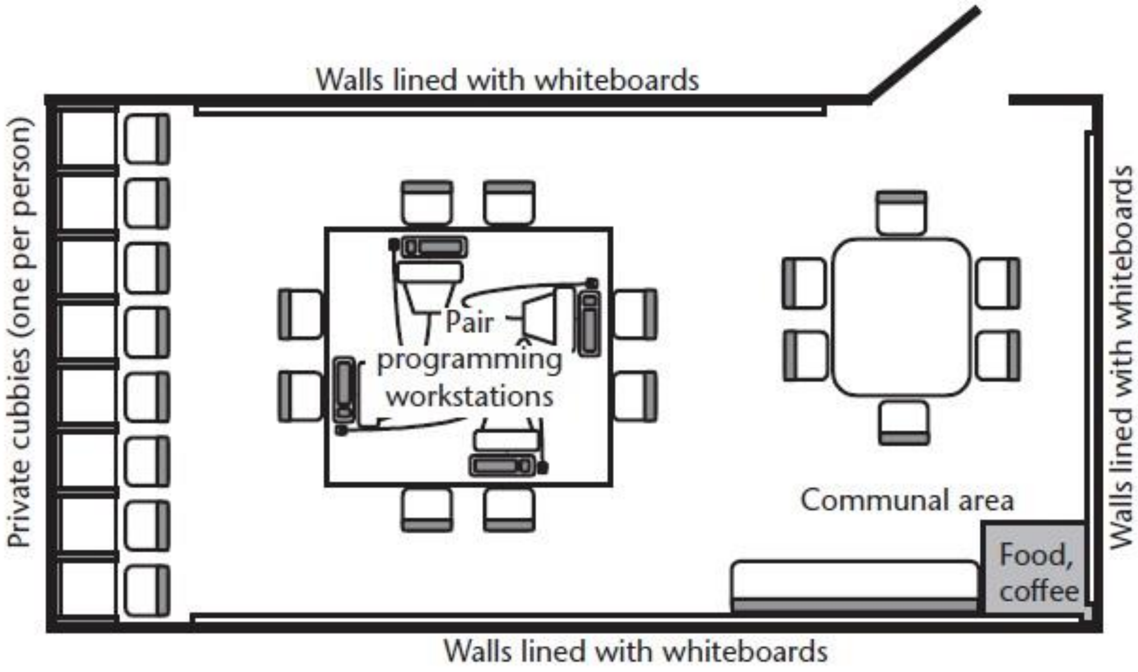

Walls lined with whiteboards
Private cubbies (one per person)
Pair programming workstations
Communal area
Walls lined with whiteboards
Food, coffee
Walls lined with whiteboards

Figure 2 – Extreme Programming Office Space

(taken from (Koch, 2004))

2. Pair-Programming

Programmers are usually expected to work in pairs on a code with one person writing and the other helping. This help includes giving feedback, advices, checking the code and discussing possible solutions. Blankenship et al. (2011) argues that pair programming enables knowledge sharing and increases technical excellence since sharing of experiences between the co-workers is enhanced.

3. On-site Customer

To adapt the product to customer needs, team needs to know both what customer wants and if customer is satisfied with the code written so far. This requires the team to have a continuous communication with the customer. On-site customer or product owner in XP and Scrum acts as a proxy between the customer and the team, participating in discussions and planning. Koch (2004) argues that having such a proxy with the team is for the teams advantage since team members do not have to make assumptions on what needs to be done.

They can ask the proxy at any moment and questions regarding the customer would be answered quickly.

4. Iteration Planning

XP and Scrum methodologies plan which features to deliver, how much resource is required to deliver them and who should deliver them before each iteration. Planning meetings require all the stakeholders to be present so that "questions should [can] be raised and disagreements settled, while all of the key people are together in one room."(Koch, 2004). Adding or removing features to/from the product is done during these meetings. Therefore this activity which is done several times throughout the project enables the team to adapt the final product to changes in requirements.

5. Daily Meetings

Collective mind-set in agile projects require every team member to know about other team members' and project's progress. This is done by having small daily meetings between the team members. Blankenship *et al.* (2011) describes several key activities in these meetings, which are

- Discussion of yesterdays, and today's prospective problems,
- Progress update from every member in the team

These activities help the team to stay up-to-date with project progress(Cervone, 2011) and realise the visible progress over the project tasks. Moreover, they can comment and feedback on people's problems in a timely manner. As a general rule, this is done in an open space, with everyone standing and for a maximum of 15 minutes.

6. Collective Ownership

Collective ownership of code is an XP practice. Every programmer in the team is "empowered to take any actions that they agree is necessary to reach the desired result"(Koch, 2004). This means that there is no ownership on any chunk of code that is written and any member is responsible for development and functionality of it as long as changes does not cause existing tests to fail.

7. Participation of Team in Decision Making

During planning and meetings, “all team members should jointly share decision authority, rather than a centralized decision structure”(Hoegl and Parboteeah, 2006, cited in Moe, Dingsøyr, & Røyrvik, 2009). This method, if the team is mature and knowledgeable enough, “let[s] those who are the closest to the object of discussion make faster decisions”(Stober & Hansmann, 2010).

8. 40-hour Week Rule

This rule, which is special to XP method limits the team’s per-man hours in a week to a maximum of 40 hours. While it does not forbid team members to work extra if they want, this rule is to maintain energy and motivation of the team members until the end of the project.

9. Small Releases and Continuous Integration

This rule, which is special to XP method states that there should be frequent releases, each one adding more value to the product towards client requirements. Moreover, “code for each story [code item] is integrated into the evolving system as soon as it is ready”(Koch, 2004).

10. Sprint

This practice special to Scrum method divides the project into smaller time-constrained chunks (usually 3-4 weeks) which produces a working prototype of the final product. Each sprint starts with sprint planning where the features of the product that will be implemented are decided. Therefore, team starts with a goal to be achieved at the end of the sprint. However, it is team’s responsibility to decide on the strategy to achieve it. Members are given freedom to choose which items they wish to implement. “Scrum’s time-boxed increments provide a mechanism for all project stakeholders to learn about the system being built on a regular basis”(Koch, 2004). At the end of sprints, a sprint review is done to evaluate what has been done during the last sprint. All the stakeholders in the project takes part in sprint reviews.

11. Product Backlog

Product backlog is an artefact special to Scrum methodology. It contains all the features that would be implemented in the project. New features can be added or removed to the backlog and it is maintained by the product owner. At the beginning of each sprint, the team decides on which features to implement from the backlog. It can be considered as a list of all the features the client wishes to see in the final product.

12. Test Oriented Design

XP methodology advices teams to produce the tests for the code before the code is written. This helps programmers to have a deeper understanding and “special insights into the code they are about to write that naturally results in better code that is freer of defects in the first place”(Koch, 2004). Moreover, this practice ensures that the test code would progress together with the product code.

Summary

Agile software development increases client satisfaction by making the process more adaptable to changes in requirements and resources, increasing stakeholder commitment, empowering teams and increasing simplicity. There are several agile methods that are designed to achieve these goals. Their common values are the agile manifesto and principles behind it. Different methods introduce same or different practices to proceed towards the best product possible by the resources at hand.

2.2 Team Performance

What is a Team?

It is required to make a clear definition of team to understand and evaluate team dynamics and team performance. Katzenbach and Smith (2005) argues that the term “team” is used without knowing the difference between working groups and teams, which prevents application of the discipline that leads to good performance.

A common definition of a team is “a small number of people with complementary skills who are committed to a common purpose, set of performance goals, and approach for which they hold themselves mutually accountable”(Katzenbach & Smith, 2005). Teams are separated from working groups fundamentally by team members focusing on the team goals rather than their individual goals. People forming the team combine their skills to “achieve something beyond the capabilities of individuals working alone”(Marks et al., 2001).

2.2.1 Teamwork

This section will briefly describe the term teamwork and qualities of teamwork under the context of agile software development. Points where agile methods differ from traditional methods will also be mentioned.

Teamwork can be described as a process of social interactions and actions. It takes a goal as an input and results in a virtual or physical product. Quality of social interactions have direct impact over the cost, delivery time, quality of the output and stakeholder satisfaction. “The concept of teamwork carries with it a set of values that encourage listening and responding constructively to views expressed by others, giving others the benefit of doubt, providing support, and recognizing the interest and achievements of others”((Katzenbach & Smith, 2005), cited in (Moe et al., 2010)).

Teams’ performance does not add up linearly with each team member. People’s performances are highly dependent on their environment(human and material). Therefore, the desired skillset cannot be obtained by adding up people having those skills. Teams are formed with people who can work together and who increase (or do not decrease) each other’s’ performance. It is not uncommon for managers to consider team members’ Belbin characteristics and form a balanced team.

In addition to considerations while forming the teams, performance of teams also depend on the monitoring and control activities over the team. Traditional project management processes rely on the proven mechanisms of planning, monitoring and control of project manager to get most from the team. Moreover, decision making is more concentrated at project manager and teams are managed to decrease conflicts and increase communication.

Teamwork is a major part of agile software development process. http://agilemanifesto.org/ (2001) states the importance of interactions and people over processes and tools. Flexibility and adaptability of these processes are results of continuous interaction between stakeholders.

Linear methods like *waterfall* divide the project execution process into stages and allocate certain responsibilities to different team members. Groups or individuals take products of previous stage (e.g. high-level design) and pass on their products (e.g. low-level design) to the next stage (e.g. testing), therefore maintaining a systematic chain.

Incremental and iterative methods like Scrum or XP “divide the implementation phase into several pieces and repeats it several times on subsets of the overall project”(Stober & Hansmann, 2010). Team members work together to release a testable prototype of the final product in iterations, building onto the previous iteration’s output. This repeated cycle enables continuous testing of the product in each iteration and provides the opportunity to change or improve the requirements with feedback from the previous product, therefore making the process more flexible to react to changing requirements. This process requires all of the team members to be active and communicating throughout the process without waiting for their turn. In this *rugby approach*, “team members work together from start to finish.”(Takeuchi & Nonaka, 1986)

Agile methods, instead of micro-managing the team, allows teams to decide for themselves. “This philosophy is a movement away from traditional command-and-control management and toward one that counts the team as an entity that has its own knowledge, perspective, motivation, and expertise”(Koch, 2004). Empowering the team with control over their own actions require involvement of team members not only by workforce but also by expressing ideas, contributing in making decisions and sharing ownership over the project.

2.3 Factors Affecting Team Performance

In previous sections, basic understanding of software project management, agile methods and team dynamics were discussed. Under the light of these knowledge, this section will focus on identification of discussed barriers in literature and other factors that were not explicitly classified as barrier but interpreted to have negative effect on team performance. Academic and published materials were taken as a basis for review. Books, case studies and journal papers were studied.

"… the use of teams does not always result in success for the organization"(Moe, Dingsøyr, & Røyrvik).

The section is divided into three distinct categories. Factors about team interaction and personal capabilities, material factors and process based/ organisational factors. Categorisation of the factors were done so that they cover the key concepts of software project management discussed in Section 2.1.1.

An overview of the main topics can be seen in

Figure 3. Evidences of possible barriers that are found in literature research will be explained in the following sections.

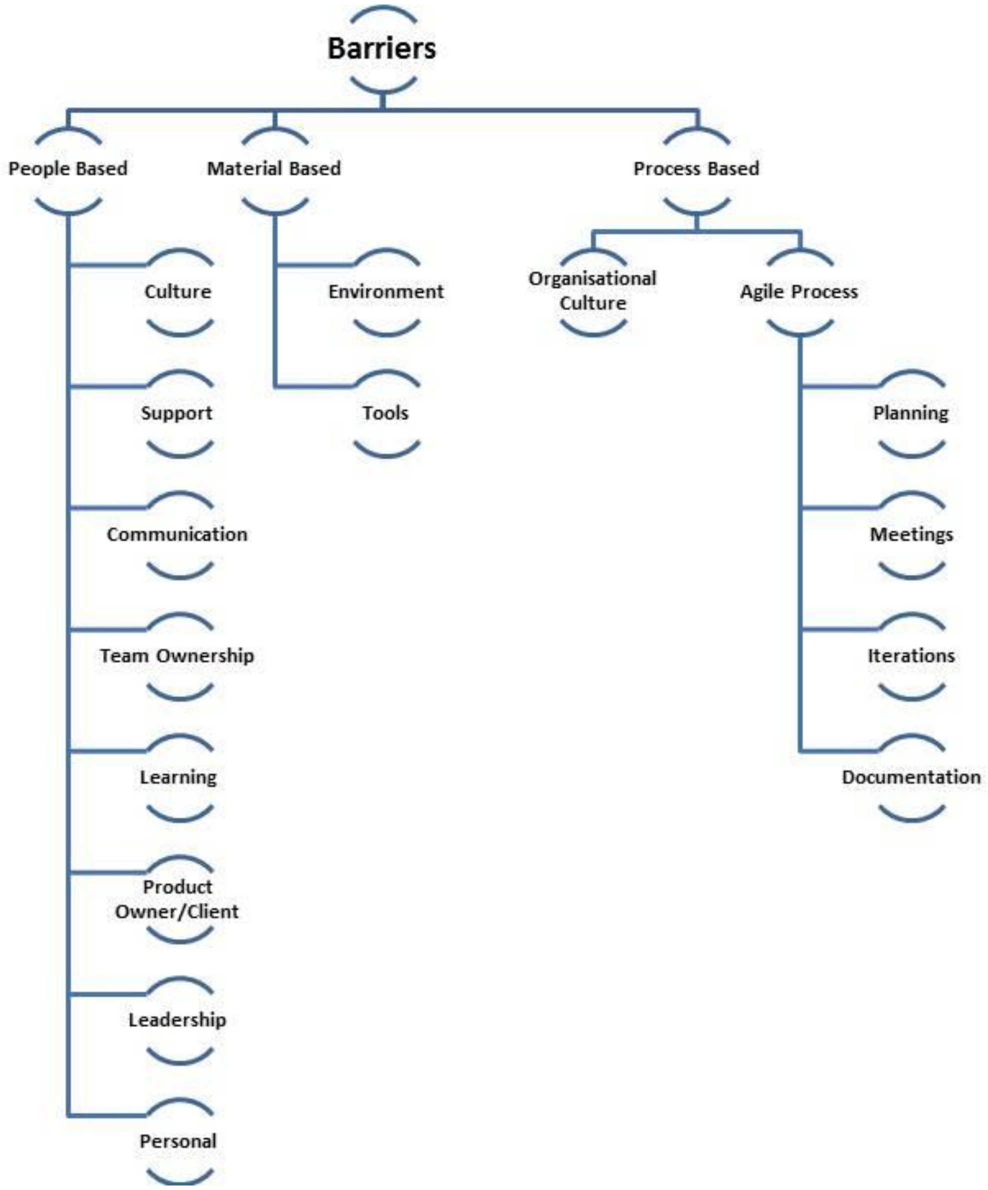


Figure 3 – Categories of Barriers to be Researched

2.3.1 People Perspective: Personal, Interpersonal and Team Interaction

Agile methods are based on continuous communication in the team and responding faster to changes in requirements. Dispersion of knowledge and information between people, relationship of team members, team characteristics and leadership in the team plays an important role in shaping the performance of teams.

Another angle to look for the factors affecting the team performance from people's viewpoint is the personalities and capabilities of the team members. In addition to certain capabilities that are required from software development teams, agile teams require their members to have extra capabilities that would fit them into the self-managing, collaborative and cross-functional team environment.

This section will review the literature to identify factors that pose a barrier against team performance under inter-related categories of self-management and personality communication, support, trust, team alignment, ownership and coordination, learning and leadership.

> *Managing people challenges is more of an art than a science; the problems' source could be the organization, the project, the team, or an individual.*
>
> *(Conboy et al., 2011)*

Self-Management

Takeuchi and Nonaka (1986) stated that in their research, one of the characteristics that leading companies show in managing their new product development process is usage of self-organizing project teams. Agile teams advance through the product development stages, starting and finalizing their work together. This 'iterative and incremental' development process is different from traditional methods where teams work like in a relay race. Instead of single or less number of deadlines with longer durations, agile teams are under pressure of many deadlines with shorter durations to work in each iteration. Therefore, team members should have a different approach coordinating their work with respect to the project. They should be efficient in organising their agendas to cope with the pace of the team.

A requirement for self-management is the absence of micromanagement in the team. Team members should be given a degree of freedom in organizing their agendas. Takeuchi and Nonaka (1986) states that involvement of upper management over the employees should be limited to providing guidance, money and support. That would allow employees to feel ownership of their own tasks and create room for innovation in their work. Moreover, it is stated in (Sanjiv Augustine *et al.*, 2005) that skilled professionals do not adapt well to micromanagement. Therefore, team members' lack of control over their work may cause them to be less efficient in creative thinking and innovation.

Communication

Information exchange in the team occurs mostly by means of verbal interactions between the team members. Communication is the basis for higher levels of interaction like support, leadership or learning. "Good communication, as a foundation to structure, requires an emphasis and value in both the human element and an understanding of roles and communication."(Fernandez & Fernandez, 2008)

Communication can be described as a function of context, medium, sender and receiver. Context being complex or simple, explicit or tacit, technical or non-technical, can have different characteristics. Medium is the means the information is being transferred such as face-to-face verbal exchange, written documents, teleconference etc. Sender and receiver are the source and the receptor of the information and maybe individuals or a group of people which are related or unrelated.

This study is focused on the barriers related to exchange of information between team members. Nature of the information can vary between technical subjects and personal matters. Literature research indicates that communication has a central role in effectiveness of teams and progress of projects.

In their research Sanjiv Augustine *et al.* (2005) stated that there is a correlation between richness of the interaction between team members and their openness to exchange of information. How easy it is to exchange information and how freely team members can discuss matters related to project effects their willingness to speak up and listen. Accepting and making constructive criticism also play a crucial role in healthy communication. Although they have danger of damaging the team harmony, through conflicts and criticism, better ideas are likely to emerge. Moreover, unexpressed opposing ideas or progressing without everyone agreeing on may demotivate the team members. Katzenbach and Smith (2005) argue that constructive conflict is essential for common understanding and purpose in the team. Moreover, most of the times, conflicts are inevitable and it is up to team to make good use of it by getting a constructive resolution. “Working in teams provides an interpersonal context in which conflicts may occur and attempts to manage them are made” (Jehn, 1995, cited in Marks *et al.*, 2001).

Communication problems often have negative effects in teams. (Curtis et al., 1988, cited in Kraut & Streeter, 1995) argued that communication bottlenecks and breakdowns are common in large development projects. It can be deducted that agile software projects are as susceptible to those bottlenecks as well. Shared leadership and cumulative progress demands everyone to communicate continuously and share their progress. This promotes a better integration between parts of code and prevents the need for rework. However, a common result of communication problems is that not everyone in the team is aware of each other’s responsibility, so there is no effective monitoring. Moe *et al.* (2010) state that lack of communication and feedback leads to more rewriting and reduces team efficiency.

When teams have communication problems they are likely to experience problems with coordinating their work (Marks et al. cited in (Moe et al., 2010))

Scrum methodology promotes colocation of the team members to increase communication in the team. Although teleconferences or mails greatly improved the communication potential of people who are not working side-by-side, face-to-face verbal communication still is the most effective way of information transfer within people. de Sousa and Almeida (2011) states that face-to-face communication creates a great potential for effective understanding. Colocation also provides the team to be at the same state of mind since information flow is more rapid. Likewise, by the result of not collocating the team members or other factors that prevents the team from communicating effectively, team members might have different states of mind about the progress of the project. Levesque et al. (2001) argue that decrease in shared mental models in teams are related to decrease in interaction.

Support

In addition to monitoring, effective communication promotes backup behaviour of team members. Agile methodologies promote members to know of each other's progress. This enables members to assist, advise or act as a backup for a member. Marks, 2001, cited in Moe, Dingsøyr, and Røyrvik (2009) describes three means of backup behaviour: providing verbal feedback or coaching, assisting behaviourally in carrying out a task and complete a task for the team member when needed. "If teammates are not looking out for, or willing to help out each other, the team will fail when any one member fails"(Marks *et al.*, 2001).

"If backup is to occur effectively, teammates need to be informed of each other's' work in order to identify what type of assistance is required at a particular time"(Marks, 2001, cited in Moe, Dingsøyr, & Røyrvik, 2009). Moreover, this shared responsibility behaviour prevents people from getting isolated in their work. Therefore, decrease in communication bears the risk of having specialized members which carry the risk of delays if specialized people are unable to perform. In their research, Levesque *et al.* (2001) found out that there was a correspondence between the time spent communicating between team members and project members' roles becoming increasingly specialized. Moe *et al.* (2010) also found out that team members not monitoring each other resulted in little feedback and almost no backup.

Trust

Collaboration is essential part of a teamwork and it requires a certain level of trust between team members. Effective communication and support requires trust of knowledge and goodwill between team members. "Without sufficient trust, team members will extend time and energy protecting, checking and inspecting each other as opposed to collaborating to provide value-added ideas" (Salas et al., 2005, cited in Moe *et al.*, 2010).

Agile teams depend largely on collaboration(see **Stakeholder collaboration** in Principles section of agile methodologies). Therefore, trust between team members play an important role in enabling the team to perform efficiently. However, it is not always the case that team is formed with people who have previous experience of working with each other. In those cases, instead of building trust over time, team members have no choice but to trust each other blindly from the beginning. Members need to put effort to create this environment and keep it that way.

Lack of trust between members may cause members to hide information, blame each other, have disputes and decrease in their motivation.

Team Alignment, Ownership and Coordination

Agile methodologies depend largely on team collaboration Involvement of every team member in decision making process, helping each other, improvement of team progress and similar traits are expected from agile teams. Unlike traditional methods where there is a certain hierarchy and command-and-control behaviour, agile teams are expected to have a more flat organisation where self-organising teams carry out the role of leader themselves.

Self-organising characteristic of teams require members to know their responsibilities as a team member, contribute and flow with the rhythm of the team without isolating themselves. Katzenbach, 1993, cited in Moe *et al.* (2010) states that successful teams gave themselves time to learn to be a team. Transformation to a self-organising team takes effort and time to learn and adapt especially for those who have no previous experience in working in such teams. Moreover, this characteristic is required to disperse to member's personal working behaviours. Team members are expected to successfully self-organise their priorities time without the need for a strong control mechanism. This deductive approach is explained by Takeuchi and Nonaka (1986) as "the individual's rhythm and the groups' rhythm begin to overlap, creating a whole new pulse."

Self-managing team members need a high-level of coordination between themselves to work efficiently. Kraut and Streeter, 1995, cited in Moe *et al.* (2010) stated that there is a significant relation between the team performance and effectiveness of teamwork coordination. Members are expected to be oriented with the team's pace, motives and working style. "High team orientation is claimed to improve the overall team performance in self-managing teams, makes team members coordinate more…" (Salas et al., 2005 cited in Gulliksen Stray et al., 2011). Individual skills should be used efficiently and should be fit into the team's goals. "Agreeing on the specifics of work and how they fit together to integrate individual skills and advance team performance lies at the heart of shaping a common approach" (Katzenbach & Smith, 2005).

Low level of coordination in the team has the risk of leaving people out of pace with respect to team, which in turn causes project progress to lag. Moreover, it would prove harder to get the best from the uncoordinated team members where there is no significant command-and-control mechanism by an authority. As the project goes more complicated and detailed, losing the rhythm of the team may decrease the team performance in accomplishing goals.

"Shared mental models are 'knowledge structures held by members of a team that enable them to form accurate explanations and expectations for the task, and in turn, to coordinate their actions and adapt their behaviour to demands of the task and other team members'" (Cannon-Bowers et al., 1993, cited in Levesque *et al.*, 2001). Forming and sustaining the shared mental model in the team provides it with high coordination in the project progress, helps team members to commit themselves in the project decision making process and project execution process, own the project and help the team to perform better. Agile methodologies give responsibility and ownership of the final product to all of the team members by integrating them throughout the whole project. Levesque et al., 2001, cited in Moe *et al.* (2010) states that for software teams, all of the team members share responsibility for the end product and for this they should develop a shared mental model by negotiating and understanding the teamwork and the task at hand.

Katzenbach and Smith (2005) argues that best teams invest a tremendous amount of time and effort exploring, shaping and agreeing on a purpose that belongs to them both collectively and individually. Acceptance of purpose in individual and group level depends on the characteristics of the team members and the level of understanding and coordination between them. Levesque *et al.* (2001) stated that members with different individual mental models have hard time coordinating their activities in the team. Salas et al., 2005 cited in Moe, Dingsøyr, and Røyrvik (2009) describes team orientation as giving priority to team goals over individual goals. Since sharing the same mental model in the team requires members to agree on a common goal, teams with higher orientation are likely to have a stronger shared mental model. Levesque *et al.* (2001) states that for temporary groups, developing a shared mental model may result in loss in time and may lead to ineffective teams. Agile teams, being temporary and under pressure of deadlines are susceptible to this kind of deficiency.

Teams that have hard time creating shared mental models may find themselves unable to make a collective decision. Moreover, as team members lose the sense of common goals, they may start to prioritise their individual goals over the team goals and progress of the project may diverge from the goal.

Learning

An advantage of agile methodologies is that it provides a good environment for team members to learn as they progress by use of constant communication and participation. Concepts like pair programming, daily meetings, monitoring and support behaviour helps the members to learn technical knowledge as well as the project progress. (Lynn, 1999, cited in Gulliksen Stray *et al.*, 2011) states that learning has a direct impact on cycle time and product success. Moreover, a coordinated team requires each member to have the same state of mind about the project. “Team orientation requires that the team members know what the project plans and goals are”(Gulliksen Stray *et al.*, 2011).

Leadership

Agile methodologies support propagating activities that traditional methodologies appoint to a single person to the team itself. Nevertheless, like every project team, agile teams need a single person to track and control everyday tasks, lead the discussions, manage the team and act as a gate between the client and the project team. Agile team leaders’ behaviours are oriented towards sustaining the team coordination, protecting the team from external disruptions and observing the team from a higher level to manage its effectiveness as a self-organising agile team. Sanjiv Augustine *et al.* (2005) defines leaders’ roles as identifying and analysing the practices that are not being followed and removing the obstacles to their implementation.

Unlike traditional boss type leadership behaviour, agile leaders are expected to be the flag carrier of the team and they should act as a role model. Burke et al. (2006) found out that use of transformational leadership behaviours are positively related to perceived team effectiveness.

Shared leadership in the team is another point of view when considering leadership in the team. Team members are expected to lead the team by their experience and knowledge from time to time. Hoegl and Parboteeah, 2006, cited in Moe, Dingsøyr, and Røyrvik (2009) argue that all the team members should jointly share decision authority instead of a single controlling leader or everyone making decisions for their individual goals independent of others.

Association for Project Management (2006) encourages project managers to demonstrate leadership that the team can follow. However, Katzenbach and Smith (2005) states that teams share leadership roles while working groups have a strong and clearly focused leader. This difference in perception of leadership is because agile teams depend on participation of team members in the decision making process. While studying an under-efficient software development team, Moe *et al.* (2010) found out that only a few team-members participated in the decision-making, and the Scrum master [leader] focused more on command-and-control than providing direction and support for other team members. This behaviour is usually the result of project managers trying to use traditional approaches as discussed in (Sanjiv Augustine *et al.*, 2005).

(Marks *et al.*, 2001) state that no work-related tasks are performed in a vacuum. There are always external distractions to the project team that may disrupt and delay them. Protection of team against such disruptions and maintaining the team's pace is up to leader. Gulliksen Stray *et al.* (2011) examined that in such a case, the leader collocated and isolated the team to protect them.

Team's efficiency with respect to the shared leadership and leadership behaviour of the team leader are closely related to each other. Team leader being not capable of performing an agile leadership may cause the team to get disoriented and disturbed by external distractions. Moreover, without the effective implementation of shared leadership, individual goals of members can gain primacy and team's pace towards the goal may slow down.

Personality

In addition to technical and agile competencies, personality of the team members in agile teams plays a significant role in the performance of those teams. Every individual in the team has different personality factors. Acuña *et al.* (2009) argue that those personality factors determine personal preferences, opinions, attitudes, values and characteristics which makes every individual different from each other.

(Young et al., 2005) made a study over an XP team to identify the appreciated personality characteristics expected from different roles in the team. Results obtained from the "good team member" category shows the personality characteristics that play more important role. Dominant characteristics of that analysis are analytical thinker, good communicator, learner and willingness to share. Likewise, "bad team member" characteristics turned out to be willingness to be dictatorial and preferring leadership over creativity.

Although different characters of individuals may prove worthy in different teams, people working with agile methodologies prefer working with people who are more inclined to team-working and self-management. Since personality of individuals play important role in their interaction with other team members, it may have direct impact on the performance of the team.

2.3.2 Material, Tool and Environmental Factors

After interpersonal and personal perspective, second angle of the factors that can create barriers against team efficiency in agile projects are material factors. Software development relies heavily on technological tools. Hardware and software tools as well as visual office tools are commonly used in agile projects. Moreover, office environments in which the teams work are proved to have significant effect on teams' performance. This section will review the literature to identify factors that pose a barrier against team performance under categories of materials, tools and environment

Materials, Tools

One of the most suggested tools in agile projects are dashboards and whiteboards. In addition to their simplicity, their high visibility when placed to a convenient place is higher than those of software tools. Nevertheless, there are various software tools in the market to manage agile projects. Teams can configure these tools to fit their processes and collaborate easily on a virtual environment. Moreover, software tools are a must for geographically distributed team members.

Marks *et al.* (2001) states that effective teams rely heavily on technology. Many software developers prefer using more than one computer and/or monitor in their workplace. In addition, some tasks or programs may require high-speed computers or internet connection.

Usage and selection of tools may have negative effect on team performance in agile teams. If a team is forced to modify their familiar working process to a new software tool, they may sacrifice their productivity. Furthermore, inadequate hardware or software tools can slow down the team's product development speed.

Environment

Agile methodologies require team members to be in continuous and intense communication throughout the project. For this reason teams are located in the same office unless they have to work geographically away from each other. de Sousa and Almeida (2011) states that face-to-face communication creates a great potential for understanding each other.

To increase the communication potential of team members and isolate them from external distractions, project leaders usually place the team in an isolated office for the project. This helps the members to work in close proximity with each other and have a sense that they are working for only that project which in turn increase their commitment. Takeuchi and Nonaka (1986) argues that all the information is shared easily without even trying when the members are located in the same place. In addition to that, de Sousa and Almeida (2011) states that frequency of the interaction between team members are closely related to the proximity between the physical spaces between them.

Inadequate working environment can decrease the team's performance through the project. Lacking facilities like large workspaces, meeting rooms, silent rooms etc. can decrease the team's interaction opportunities and get them distracted while working.

2.3.3 Process Based and Organisational Factors

Organisational culture, procedures and how processes are handled are deterministic in how teams perform. This section will review the literature for barriers resulting from organisational culture and agile project process.

Organizational Culture

All the teams in an organisation act according to the culture of the organisation and reflect the culture of the mind-set it belongs to. Therefore, effectiveness of the teams is largely dependent on behaviour and agile understanding of the organisation and it is important for organisations to support the agility of the agile teams performing under their roofs. "Teams themselves can influence the internal organization of teams, but team performance depends not only on the competence of the team itself in managing and executing its work; it also depends on the organizational context provided by management"(Guzzo and Dickson, 1996, cited in Gulliksen Stray *et al.*, 2011).

Organizational culture can enhance the performance of teams as well as hinder them. Procedures, hierarchical bureaucracy and traditional mind-set can influence team members not to act according to how their agile method requires.

Agile Process

All the agile software teams act and decide according to some pre-defined set of rules and with the help of standardized artefacts. Therefore team's success in application of their specific set of process rules have significant effect over the performance of the team. "Without a proper development process in place, a project team could operate in a chaotic manner, resulting in low productivity and poor system quality." (Paluk et al., 1993, cited in Chiang & Mookerjee, 2004). Although agile methodologies are most effective when oriented on "product instead of "process" (Fernandez & Fernandez, 2008), sticking to rules of process acts as a steering wheel for the team to reach their goals. This section focuses on subjects related to planning, meetings, iterations and documentation.

Planning

Planning is one of the most important activity in agile software projects like any other project type. Agile projects differ from traditional ones with simpler and repeated planning that enables teams to take corrective actions in time. Marks *et al.* (2001) stated that teams primarily focus on planning to accomplish their objectives. Planning activities involve pre-action preparations which can be identified as “prioritization of goals and sub goals for mission accomplishment”, “…identification of main tasks as well as the operative environmental conditions and team resources available” and “…development of alternative courses of action for mission accomplishment… discussion of expectations, … prioritization of role assignment, and the communication of plans to all team members.”(Marks *et al.*, 2001), post-action critics to “better understand the underlying causes of previous performance [to] … better prepare for future efforts” (Blickensderfer, Cannon-Bowers and Salas, 1997, cited in Marks *et al.*, 2001).

Project context going out of the plan scope, team (willingly or unwillingly) not acting according to plan or being unable to accomplish the iteration plan is not uncommon in projects. “No work-related tasks are performed in a vacuum, unaffected by deadlines, time limits, or schedules.”(Marks *et al.*, 2001) Not acting according to plan have many risks. Gulliksen Stray *et al.* (2011) observed that team members prioritized individual goals over team goals and motivation of the team members decreased when an unrealistic plan is made.

Meetings

An important requirement for every team to stick to in the agile process is meetings. Kraut and Streeter (1995) defines meetings as a way of formal information exchange. In addition to periodic meetings with bigger contexts such as iteration planning and retrospective meetings, daily stand-up meetings play an important role in synchronizing the team members and get rapid feedback. General meetings help the team set their long-term objectives while daily meetings help them to make fine-adjustments to their daily actions.

Unsuccessful realization of such an important element of projects may decrease teams’ overall performance. Some of the reasons for such results are participants being unprepared, necessary decision makers and team members not being available at the meetings and context of the meetings being out of scope. Moreover, Gulliksen Stray *et al.* (2011) stated that the team they observed had problems with not spending enough time planning the upcoming sprint.

Iterations

Agile methodologies depend on repetitions of smaller scale execution phases aimed to bring a working prototype of the final product at the end. These iterations start with the iteration planning where the objectives are determined and ends with an evaluation and retrospective meeting. As a rule, scope of iterations are fixed until the end. In other words, it is seldom that new objectives are added or current objectives are removed from that iteration’s feature list. Sprints have fixed durations which are decided at the beginning of the projects. Those durations are determined according to the complexity, time constraints and available resources of the project. “For a given set of requirements, a shorter construction time should translate to a larger team and thus more expensive coordination among the team members.” (Chiang & Mookerjee, 2004)

Moe, Dingsøyr, and Røyrvik (2009) observed that making changes during the iterations resulted in difficulties with delivering according to the sprint-plan. Chiang and Mookerjee (2004) states that unrealizable planning causes team to exhaust and increases stress during iterations.

3 Methodology & Data Collection

3.1 Introduction

The purpose of this chapter is to describe the research, data collection and data analysis strategy. Chapter begins by defining the research question and describes the methods of the research by explaining the rationale behind the selected methods. Next, research procedure, which includes academic review of knowledge and data collection and analysis regarding the research question is defined. Chapter finishes by explaining the ethical considerations during the research and limitations of the research method.

3.2 Research Question

Based on initial research about agile software project management and team dynamics, author decided that factors relating to the performance of agile teams are important when managing the well-being of the projects. Assuming that eliminating negative effects are as important as fostering positive ones, this research is trying to find answers to following question:

What barriers are there that decrease team performance in execution of agile software projects?

3.3 Characteristics of Research

3.3.1 Introduction

This section explains the characteristics of this research method to gives the reader a better understanding of what methods researcher used to create the knowledge and the reason for selection of the methods. It includes the research paradigm of the researcher while conducting the research, then continues with explaining the methods of data collection to be used.

3.3.2 Interpretivism (Research Paradigm)

Understanding and evaluating a research question is affected by the view of researcher’s point of view. Moreover, the justification of a theory in a research depends on the rationale behind the explanation and proof of the data and information given by the researcher. Therefore, the knowledge created in a research is explained from the researcher's point of view. Fellows and Liu (2009) state that scientific knowledge is falsifiable rather than verifiable, meaning that while result of a single test to disprove the theory is sufficient, hundreds of results that supports the theory is not enough to prove its correctness. This statement creates the need for the researcher to explain his point of view. Especially in subjects related to people's behaviour like this thesis' topic, results depend significantly on how researcher approached the question and what he considers to be important .

Saunders et al. (2009) and Fellows and Liu (2009) mention the term paradigm in their work. "Paradigm is a way of examining social phenomena from which particular understandings of these phenomena can be gained and explanations attempted" (Saunders *et al.*, 2009). Fellows and Liu (2009) defines it as a framework to view a system, comparing it to a lens. Author's paradigm in this research is connected to how he tries to understand the sources of barriers and whether these are facts independent of social environment or not.

Fellows and Liu (2009) describe two different paradigms in their research: Interpretivism and positivism. In addition, Saunders *et al.* (2009) mentions about two other paradigms: Realism and pragmatism. Author of this research's paradigm is interpretivism. It will be explained and rationalised together with the alternatives.

Interpretivism

This research adopts interpretivism for its research philosophy. It tries to identify various barriers that decrease team performance without assessing their importance or level of impact with respect to each other. Therefore, for this exploratory research, it is important "to understand differences between humans in our role as social actors"(Saunders *et al.*, 2009), and avoid generalizations. Because, as it was stated in (Fellows & Liu, 2009), one person's reality that is derived by observations and perceptions can be different from another's. Therefore, author believes that results found are in this research dependent on various factors like culture, time-frame and personal opinions of research participants and can be different when the study is repeated under different conditions.

Alternatives

There are three alternatives mentioned in works of (Fellows & Liu, 2009) and (Saunders *et al.*, 2009). These are positivism, realism and pragmatism. This section will explain those alternatives and will give the rationale for not adopting those paradigms for this research.

According to (Fellows & Liu, 2009), positivism is an approach that recognises only non-metaphysical facts and observable phenomena. Payne and Payne (2004) states that in positivism, the researcher makes up theories that are external to observer, meaning that the interpretation does not depend on perceptions of individuals. Those theories are testable and repeatable and come to form from deductive reasoning.

(Saunders *et al.*, 2009) states that in realism approach, what our senses show us as reality is the truth. Considering interpretations of (Saunders *et al.*, 2009) and (Payne & Payne, 2004) key difference between positivism and realism is that while they both assume a scientific approach to create knowledge, realism accepts that there are less observable forces that lie behind the phenomena. In that sense, realism can be thought as an approach accepting that observable phenomena are external to human existence but giving less emphasis on empirical understanding with respect to positivism.

Pragmatism is another approach that is mentioned in (Saunders *et al.*, 2009). The logic behind this philosophy is that instead of selecting an absolute stance for the research like positivism or interpretivism, researcher can adopt the most appropriate approach that enables him to explain the phenomena best. Saunders *et al.* (2009) states that in pragmatism, the important thing is to answer the research question and different philosophies may be beneficial for answering some questions in the research.

These three different alternatives to interpretivism were not adopted for this research since the research question and author's approach to creating knowledge does not require absolute facts and statistical derivations. Moreover, aim of the study is to identify the barriers, which are interpretations of the research participants. Therefore, a philosophy that accepts that the knowledge that is trying to be created is subjective and dependent on the stance of people who interprets them is adopted for this research.

3.3.3 Induction (Data and Theory Relationship)

Induction and deduction are two methods to create a relationship between the collected data and formulation of an hypothesis. It is claimed in several studies (Fellows & Liu, 2009; Saunders *et al.*, 2009; Strauss & Corbin, 1998) that deduction is used when testing a hypothesis using the collected data and when application of empirical methods are suitable. However, induction method uses the collected data to arrive to generalisations and theory building.

This research, as it adopted interpretivism approach and theory building with the collected data, uses induction to create knowledge. Studies mentioned earlier state that literature review and data should be analysed together when using inductive approach. This research critically analyses the data collected from the interviews with respect to the literature review to create knowledge.

3.3.4 Qualitative Inquiry (Data Collection Approach)

Method of inquiry explains the selection of how data will be collected to explain and prove/disprove the research question. Studies made by (Fellows & Liu, 2009; Payne & Payne, 2004; Saunders *et al.*, 2009) state that data collection methods can be qualitative, quantitative or a mix of the two. Qualitative methods are used when the concern is to find out the interpretations of people with smaller sets of samples. Quantitative methods are used when statistical explanations, behaviours that can be represented by numbers, associations between variables are studied with a bigger sample set to permit a level of generalisation.

Selected approach for this research is qualitative inquiry. Data to be collected is required to be interpretations of people who are knowledgeable or experienced enough to express their point of view with respect to the research question. No statistical derivations are intended to be made. Moreover, the sample set is designed to be small enough to manage in a limited time-frame, large enough to cover different viewpoints.

3.3.5 Interviews (Data Collection Method)

To collect relevant data for qualitative research, two common methods that can be used alone or together are observation and interviewing. For this research, data was collected with in-depth interviews. An alternative for this research would have been examining documents and recordings that were relevant to the research question. However, this method was not used for lack of resources and possibility of researcher bias while analysing the materials.

3.4 Research Procedure

This section explains the steps of the overall research from beginning to the end. 3.4.1-Literature Review explains the approach in selection and inspection of knowledge in literature. 3.4.2-Data Collection explains the selection of interviewees and their characteristics relevant for research. Moreover, interview process is explained. 3.4.3-Data Analysis explains how literature knowledge and data collected are merged together to answer the 3.2-Research Question.

3.4.1 Literature Review

The approach to literature was from two angles. Firstly, books about agile project management were studied to get a solid understanding of the context of agile project management. Next, academic journals that were written about agile project management were studied. Articles can be grouped under the following subjects:

- Agile methods versus traditional methods,
- Case studies about efficiency of agile project groups,
- Development of agile theory,
- People oriented challenges in agile projects,
- General Software failures.

Following, using references obtained from previously read articles and new searches, the subject of teamwork was studied. Articles can be grouped under the following subjects:

- Teams versus working groups: What makes teams functional
- Case studies of teamwork in agile projects,
- Productivity in software teams,
- Leadership in teams,
- Factors affecting team functionality, efficiency and progress.

Using the literature knowledge, specified barriers were listed. Moreover, possible barriers that were interpreted by the author but that were not specified explicitly were added to the list. Next, these barriers are categorized under logical topics.

Chalmers University Library and Google Books were used for available academic books. Chalmers Internet Library, Northumbria Internet Library and Google Scholar were used for article search.

3.4.2 Data Collection

After literature review was finished, data collection work was started. The research was designed to be a qualitative one and data collection method was selected as in-depth interviews.

Participants

"The study of a small sample of subjects might be more appropriate than a large number as with the deductive approach"(Saunders *et al.*, 2009). 8 agile project managers were contacted for interviews. 3 of these participants were reached to using Charm fair 2013 in Chalmers University of Technology (http://www.charm.chalmers.se). Remaining 5 people were found by search in LinkedIn(http://www.linkedin.com). Of these participants:

- 3 were Swedish, working in Sweden, 4 were Turkish, working in Turkey and 1 was Turkish, working in Finland.
- All of the participants were working in agile software projects or had experience of agile projects.
- 7 of the participants were using Scrum methodology while one of them was using a mixed methodology of Kanban and Scrum.
- All of the participants had project management experience, mainly the scrum master role.

Interview Process

After an interview date was arranged, consent forms and interview topics were sent to participants 1-2 days before the interview. Signed consent forms were collected before the start of interview. Whole interview process took three weeks to complete.

4 participants from Sweden were interviewed face-to-face in their offices. Remaining 5 participants were interviewed over Skype. Interviews were voice recorded and notes were taken by the researcher during the interviews.

Unstructured interviews were conducted since an exploratory research was being made and researcher did not want to effect participants to have biased answers by asking direct questions or limit scope of the conversation by defining borders to questions. Appendix A – Interview Topics was used as a guideline during the interviews. Participants were asked to explain what kind of barriers they experienced or think there are about several topics.

A preliminary interview was made with a professional as a pilot. Results from pilot interview are not included in the research. The way questions were asked and distribution of topics in the guide were modified accordingly. Interview durations varied between 30 minutes and 2 hours depending on time available and amount of information participants were willing to share.

3.4.3 Data Analysis

Data analysis was done in two consecutive steps

Transcribing

After the interviews, voice recordings were listened again by the author and notes were taken.

Categorizing

After all the interviews were transcribed, answers that were about similar topics were grouped together and listed. Author interpreted and paraphrased answers instead of citing them without changing the meaning and intention of them.

3.5 Ethical Considerations

Participants of interviews were project managers and they were asked to talk about team performance. Some of the discussion topics like personal competencies or communication problems had danger of exposing team members' personal deficiencies or habits that interviewees' think they have. Exposure of these information may put interviewees in a difficult situation.

To make sure that interviewees will not be threatened in their professional and personal life, information gathered were documented anonymously, making sure that identities of interviewees will not be exposed. Moreover, interviewees were informed about those ethical considerations and were asked to confirm that they accept the interview under those conditions.

In addition, all the interviews were voice recorded with interviewees' consent. Recordings were kept in a safe environment and were listened only by the researcher. All the recordings were destroyed after the research was finished.

A copy of consent form which all participants were asked to sign can be found in Appendix B – Interview Consent Form

3.6 Scope and Limitations

Due to limitations on time and resources, research's scope was limited. However, limitations were kept at a level such that they did not prevent the research from making logical or unrealistic predictions.

Literature review was focused on two methodologies that use agile principles: Scrum and XP. Moreover, geographical and cultural variations on perceptions of team efficiency were not studied. Instead, team efficiency was assumed to be interpreted same globally.

Data collection was limited to agile project managers whose experience was mainly on Scrum methodology. Moreover, participants were either scrum masters or product owners. Lastly, participants were from countries with different business cultures.

4 Results & Data Analysis

4.1 Demographic Data

Before the interviews, personal information related to their profession was collected. Below are information about the participants:

- Age
    - Between 31-35
    - Mean: 33,8
- Gender
    - 7 male
    - 1 female
- Country of work
    - 3 in Sweden
    - 5 in Turkey
    - 1 in Finland
- Job Titles
    - Product owner
    - Project manager
    - Scrum master
    - Scrum team member
    - Principle software engineer
    - Business analyst
    - Agile consultant
    - Founding partner
    - Agile coach

Some participants had more than one job title

- Experience in working with agile projects
    - Between 1.5 years and 6 years
    - Mean:3,7 years

Their experience includes working as a team member and/or project manager

- Current Job Industry

- Automotive
- Tobacco
- Information systems
- Telecommunication
- E-commerce
- Banking
- Entertainment
- Mobile devices
- Public sector
- Transportation
- Retail
- Architecture
- Insurance
- Defence

Some of the participants worked in projects of more than one industry.

4.2 Analysis of Results

Respondents answers included negative as well as positive factors that they have observed. Only the negative points were included in this research. Similar answers were grouped together. A summary of answers were given in each section as numbered items. Numbers do not signify any ranking or importance relationship between the items.

4.2.1 Interpersonal and Personal Perspective

This sub-section lists the findings gathered from the interviews categorized under the subjects of *culture, support, leadership, communication, team-ownership, learning, product owner/client* and *personal adequacy*.

Culture

Question of “What do you think about culture’s role in team performance?” was asked to participants. Their answers included negative as well as positive factors that they have observed. Only negative points were included in this research. Some of the participants said that they did not observe any negative effect of culture based factors. Similar answers were grouped together. A summary of answers were given below as numbered items.

1. People with distinct cultural differences sometimes create social interactions that other team members feel uncomfortable. One of the participants mentioned a team member-who was from another country culture- was “speaking demanding and with a tone”. Other members in the team felt uncomfortable for getting in social interaction with that person. Another participant commented that his team was from a more relationship focused culture and the way of speaking people they communicate with for their project was unusual to them. Some team members felt offended by the direct speaking habit of these people. He added that those people were also thinking that the team members were “very emotional”.
2. Another factor that was addressed was ego of team members preventing them from being a good team player. Lack of traditional hierarchy based titles in the scrum teams offended some people when they started working with scrum. Losing their ‘senior’ titles to mere ‘scrum team member’ title made them feel uneasy. Participant commented that some people even left the company after their titles changed. Another participant noted that the ‘master’ phrase in the ‘scrum master’ title created “ego boom” in some people. They all argued that egos getting is a potential barrier in the teams.
3. One of the participants argued that significant age differences hinder people from working as a team. Some people felt uncomfortable working with other team members who were significantly younger. He argued that age differences were preventing people from having a fruitful interaction.
4. One of the points that most of the participants agreed on was that working with introvert people is a serious problem. Those team members who were used to working alone and not comfortable with high degrees of social interactions were hindering the teams’ performance. Their “reactions against socialising” were problematic when creating an effective team-working environment.
5. One of the respondents commented that there are sometimes people with “sharp characteristics” in the teams. He argued that “It can be hard to form a team culture with these people. The project manager needs to

find a midway between people to make everyone as comfortable as possible”.

6. One of the respondents commented that some people in the team are used to working ‘according to the book’ and added that they were trying to stick to the rules even if the rules were not helping.
7. One of the respondents noted that there are some people who are disciplined and some people are more slack during work times. He added that “if people who are not giving their hundred per cent cannot be addressed by the project manager, it can be a barrier”.

Support

Question of “What do you think about support’s role in team performance?” was asked to participants.

1. Some of the respondents argued that creating the environment for support is not easy. One of the reasons for this problem is that “it is often considered to be a routine and boring task”. Encouraging team members to give and request support in the team takes time. A respondent commented that “you cannot expect the results to just produce from thin air; you have to make an effort”.
2. According to respondents, trust was one of the factors that hinders the support environment. People not being honest to each other decreases the time they spend on giving and receiving feedback and help. Moreover, one respondent commented that blaming each other when there is a problem to get rid of the possible responsibility of failure causes the team to dissolve. One responded commented that creating an efficient team environment requires a certain amount of trust between the members and this requires the team members to be honest to each other.
3. Two respondents argued that teams do not usually have time for giving and receiving proper support. Team members become so occupied with their tasks at hand that they cannot respond to support requests from other members. Moreover, one responded argued that when people are new, they see problems but when they get experienced, they cannot find time to make the effort to make them right.

4. Another comment on this topic was that team members are not aware of their expertise and for that reason they do not act proactively when they could.
5. Behaviour of technically senior people in the team is important according to some of the respondents. One respondent claimed that some experienced people hide information intentionally. Another argued that experienced people prefer to work with other experienced members to advance more in their technical expertise. Moreover, one claimed that some people try to stand out in the team by trying to act like they know more than they do. This results in inefficient support and discomfort from other team members who do not appreciate the behaviour.
6. Hesitating to request help is one of the common problems according to respondents. Some team members are too shy to ask for help or they think that people judge their expertise when they do. One respondent claimed that if there is a team member who hesitates to ask for help, than the required team environment was not created. Moreover, he added that refusing to help for whatever reason discourages other team members to ask for help later on.
7. One respondent argued that support in a way of criticism instead of encouraging can be dangerous for the team.

Leadership

Question of “What do you think about leader’s role in team performance?” was asked to participants.

1. Some of the answers were related to the characteristics of people in the team and how the leader interacts with them. One respondent argued that if there is a fragile person in the team, the leader has to be careful when addressing him since they can demoralize easily. Moreover, another respondent noted that some people in the team are hard to reach and leader should pay special attention to them to make them express themselves.
2. Another point in the context of leadership is that leader’s personality has a direct effect on the performance of the team. One respondent commented that the leader should be a role model and set an example for the team. Another argued that poor leadership causes team to

underachieve and lose their motivation. Making mistakes was accepted as a normal event but one respondent said that leader should be able to admit the mistake and try to repair it. Moreover, one respondent argued that leaders with inadequate social capabilities and who cannot lead the team to act according to their designated process causes performance droppings in the team.

3. Leader’s behaviour against the team is also important in motivating the team. Two respondents argued that the boss-type hierarchical leadership does not apply to agile teams and even demotivate the team. Roles of the leader and product owner should be distinct and the leader should neither try to act like a product owner nor allow the product owner to micromanage the team members
4. Team leadership was another point that some of the respondents thought was important. One respondent argued that there should be context based natural leadership where anyone in the team who has experience about the current context should act as a leader when it is convenient. Otherwise the team would not be able to use the knowledge of the experienced people in the team. It is commonly accepted throughout the respondents who commented on team-leadership that it is more beneficial for the team to have the team itself as a hero instead of individual heroes.
5. Protecting the team from outside distractions was thought to be high of importance in the team. Sales department, customers, upper management or people from other projects constantly interrupts the team to get support from them. One respondent said that the team should be protected by the leader from outside distractions to keep them focused on their project tasks. However, one respondent argued that sometimes leader’s political force may not be enough to protect the team member from requests from someone higher in hierarchy. Nevertheless, outside distractions was accepted as a factor that hinders the team performance and it should be addressed by the leader if needed.
6. One respondent argued that the leader should be focused only on the team throughout the project. Handling several roles in several projects causes the leader to be a “part-time leader” for the team, which

decreases the team’s performance by loosening their focus or avoiding the leader to solve the problems right away.

7. Last but not the least, one respondent argued that the leader is there to solve problems of the team and “if he does not do anything when there is a problem, it affects the whole team.” He also noted that the leader should answer questions(e-mails) directed to him as quickly as possible to keep the team going otherwise there will be people who wait for an answer to continue working on a task.

Communication

Question of “What do you think about communication’s role in team performance?” was asked to participants.

1. Proximity between the team members was commented on by three of the respondents. One claimed that if the team members are seated more than 10 meters away from each other, they stop communicating. In addition, another respondent added that even when they are seated close, they sometimes do not communicate enough or they do not facilitate the right kind of communication. They agreed that the leader should facilitate the communication environment in the team. Not creating the right environment for communication and leading the people to facilitate the communication spirit causes the team to disconnect.

> *“Being kind is not always ethical”*
>
> *(Respondent)*

2. One respondent argued that some people believe that they should only communicate when there is something to report, which in turn creates an environment that hinders the collaborative environment. He also noted that team should always be in communication even if the context of the interaction is not directly related to tasks stating that “you can never communicate too much”.
3. Personal sensitivities and irresponsible actions like aggressiveness and talking behind people’s back creates lack of trust in the team which in turn hinders the communication environment. One respondent commented that the culture that especially the competitive private sector

creates push people to act self-centred and incompatible to team-working. This results in people having a difficulty in communication and causes demoralization in the team.

4. Some of the respondents added, shyness, not being pro-active and not having ambition to succeed also count as personal characteristic that hinders the communication in the team.
5. One respondent argued that the communication skills and technical skills of a team member should be in balance. Both the skilled unsociable and unskilled social characters are threat to effective and fruitful communication.
6. A factor that most of the respondents agreed on is that if one of the team member fails in the team, then the whole team fails. Although it is not directly related to communication, most of them commented about this phenomenon when they were asked about communication.
7. Not all the people are comfortable with handling conflicts and concluding a useful outcome from them. Some of the respondents argued that avoiding conflict creates more problems in the team. Some people who are not comfortable with having a conflict try to avoid it. Two respondents noted that this behaviour results in bigger conflicts or even fights. “Following the herd not to cause conflicts or be the odd one” is actually harming the quality of the product in the long run since it prevents developing of new ideas.

Team Ownership

Question of “What do you think about team ownership’s and team leadership’s role in team performance?” was asked to participants.

1. A respondent argued that the team is much more than just being in a group, that the team members have to understand their part in the team and respect the team rules. Without creating the mind-set to take responsibility and action as a team, personal benefits and prioritizations get in the way of common understanding. Conflicts arising from different personal benefits or focusing on different corners of the iron triangle (time vs. cost vs. quality) are barriers against better team ownership. One respondent claimed that people sometimes prioritize their own goals

over the team goals and this causes the team to lose its focus on common goals.

2. One respondent argued that members who are unable to self-manage their routines and tasks are unable to contribute to a self-managing team.
3. Team ownership requires working together to achieve a common goal. One respondent claimed that when there are people who have no enthusiasm to advance and improve, senior people stop spending time and effort to help develop those people.

> *“Members are responsible for their tasks and team is responsible for all the problems”*
>
> *(Respondent)*

4. A team is as strong as its weakest member. One respondent claimed that it is harder to build up a team than to tear it down. Building a team takes time, dedication, patience and hard work while one odd person is enough to tear it down. Odd people who are not for teamwork can cause the team to dissolve.
5. One respondent argued that team members should feel valuable in the team in order for them to own the team and its principles. People tend to not own the team when they are not cared inside the team. He also claimed that team should give emotional and technical support to members to help them overcome the feeling of being inadequate. People without self-trust tend to perform less efficient.

Learning

Question of “What do you think about team and individual learning’s role in team performance?” was asked to participants. Participants were asked to consider both the project and technical context of learning.

1. A respondent stated that during projects or during the employment time of the team members, individuals lose their interest in learning. This loss results in slower learning and adaptation to new technologies.
2. Organizations with high turnover rates struggle to keep and spread the knowledge inside. When people leave the team in the middle of projects, team loses knowledge. Therefore, when mechanisms of learning and

spreading knowledge are not implemented efficiently in teams working in organisations with high turnover rates, teams lose the knowledge base that they created.

3. In addition to knowledge loss in case of high turnover rate, same responded argued that when knowledge in the team is not spread to several people, when experts get sick, team loses their power for the time that person is not available.
4. It is also argued that to maintain the knowledge sharing environment, teams should be able to self-manage themselves. When the leader of the team starts to act as a traditional project manager and tries to micro-manage the team, team loses its capability to learn.
5. Conflicts arise from different interests of people or departments or when people from different companies work together. It is argued that sometimes people hide information which in turn decreases the team's learning capacity.
6. A respondent commented on hiding information stating that managers should know when to hide information and when not to. Some information can make the team nervous when they have no experience of handling that information.
7. A respondent argued that when experts are always given same kind of tasks that are related to their expertise, they cannot learn. Moreover, he argued that experts of some specific subjects tend to finish tasks of their expertise by themselves to save time from teaching others. This hinders the team from learning about what they do not know.
8. A respondent stated that sometimes team members think that everyone is knowledgeable about a topic and do not think that they should spread their knowledge about that topic. This pre-assumption may cause the team to miss important information that they should know.
9. Another problem according to one respondent is that people feel uncomfortable when they are introduced with new technologies or tools. Since people feel more comfortable working with what they already know, adapting to new ways of working creates the fear of being inadequate. Therefore, they try to avoid learning new tools and technologies.

Product Owner/Client

Question of “What do you think about product owner’s and client’s role in team performance?” was asked to participants.

1. Most of the respondents agreed that not having the product owner at close proximity to the team decreases the speed at which the team delivers tasks. It is argued that when the product owner does not spend enough time with the team or is not easily available to the team, team makes more mistakes, questions about tasks takes more time to get answered, motivation of the team decreases, team or the leader starts to take over product owner’s responsibility.
2. A respondent argued that when the team is provided with indefinite tasks, they spent most of their time trying to figure out the details of the tasks and usually waste time over unnecessary coding.
3. Another respondent argued that the product owner should lead the team instead of pushing them. When the product owner creates a bad or vague vision for the team, it is argued that team members lose their sense of ownership over the project. Moreover, if the product owner fails to explain why something has failed team cannot learn from the failure and can fail again.
4. Another common comment on product owner’s role is that behaviour of product owner directly effects the performance of the team. A respondent argued that the product owner should be educated in agile methodologies (scrum in the interview) to be capable of supporting an agile team. Another respondent claimed that some product owners focus on the process instead of the product or focus on cost when the team tries to focus on quality. This results in team losing its motivation and delivering poor quality products.
5. Moreover, two respondents claimed that authoritarian and micromanaging product owners make the team feel pressured. When product owners with this kind of personality are co-located with the team, team feels stressed.
6. When there are more than one product owners with different agendas, team can get caught in the middle of interest conflicts.

7. A respondent argued that product owner should be able to speak the same technical language with the team. When product owner lacks the knowledge about the platform the team is working with, problems with communication and understanding occurs.

Personal

Question of “What do you think about personality’s role in team performance?” was asked to participants.

1. Most of the respondents claimed that individuals should be educated and be adaptable to agile methodologies. Working with agile methodologies requires different set of social skills and mind-set. If people cannot let old habits from traditional working methodologies go, they cannot adapt agile. Lack of adequacy in agile methodologies causes individuals and therefore team to lose their pace and efficiency.
2. Two respondents claimed that people working continuously with a high pace to deliver for the iteration or for the final release feel exhausted and lose their motivation together with their efficiency. Exhausted people tend to take shortcuts and make more mistakes.
3. Adequacy in technical knowledge is another most-commented and obvious factor that effects the performance of the team. People who lack the technical knowledge to complete tasks assigned to them slows down the team.
4. In addition to technical knowledge, a respondent argued that experience of individuals is an important factor in determining the performance of the team. People who lack experience in the team tend to be less self-reliant and they prefer not to express their ideas. Those people who prefer to follow the horde instead of expressing their ideas hinders the team’s potential of creating new ideas and alternative solutions.
5. Some respondents claimed that people sometimes have their own goals or agendas in the team. Those people disturb the team ownership by not dedicating most of their effort in the teamwork and tasks assigned to teams.
6. Capability of handling stress is another point that was discussed by one of the respondents. Pressure from customer, product owner, team or deadlines may cause people to get stressed and people who cannot

effectively handle stress tend to make mistakes and decrease their performance.

7. A respondent argued that some people prefer to work alone instead of working in a team. Those people may cause the team to lose their sense of being a team and therefore hinder team ownership.
8. Career goals of people are claimed to be important by a respondent. Individuals who has distinctively different career goals decreases the team alignment.

4.2.2 Materialistic Perspective

Question of “What do you think about material environment’s and tools’ role in team performance?” was asked to participants.

Material

1. Most of the respondents argued that being physically co-located and working in close proximity with each other is important in team performance. When team members are not co-located at the same place, or are distributed inside the office, their potential for communication decreases. One respondent claimed that when they are seated more than ten meters apart, they stop communicating. Decrease in support and communication when team members are not close to each other were two common problems with co-location of team members.
2. One respondent argued that usage of software tools in project planning decrease the potential for communicating and expressing of ideas about the project progress between team members. Moreover, software tools decrease the visibility of the project progress because they are not always in front of people. The respondent claimed that physical tools such as whiteboards and post-it should be selected and used wherever possible.
3. In addition to usage of software tools two respondents commented on the selection of the right software tools. They argued that selection of the right tools are important for the project progress. All agile teams are unique and they have their own ways of working. Teams should consider different tools and select the ones that are most compatible with their

team, context and processes. Selection of wrong tools or trying to adapt the processes to the tools result in "spending more time on tools than on developing software".

4. A respondent argued that even though teams are co-located, when the environment they work in is crowded, because of the noise and "chatty" people, team members get distracted. He claimed that when possible, private rooms should be allocated to teams so that they can work without distractions when they need. However, another respondent argued that locating all the team members in a single room makes them feel like they are "zoo animals", isolated from the office environment.

> *"most employees have better hardware at home than they have at work"*
>
> *(Respondent)*

5. One respondent claimed that today, software tools require much more processing power, memory and network speed. When hardware the team members use become inadequate for the software they use, their progress may slow down.

4.2.3 Agile and Organisational Process Perspective

Organisational Culture

Question of "What do you think about organisational culture's role in team performance?" was asked to participants.

1. Most of the respondents argued that organisational culture and related mind-set has a big effect on the team's performance. A common comment was that the organisation should embrace the agile culture to be able to spread it to the teams. Agile teams operating in traditionally run organisations have hard time applying the concepts of agile development in their projects.

> *“When you go higher in the hierarchy, you see another culture. This culture is very different from what your team is trying to achieve”*
>
> *(Respondent)*

2. A respondent argued that application of hierarchical “command and control” behaviour and micromanagement from upper management to the team causes team to lose their self-managing feature and demotivates the team members. He commented as “there are emotions, feelings and passion in a team, but when you work with command and control, there are rules and orders instead”.
3. A respondent argued that problems arise when people in the organisation have to take role in more than one project at a time. Those people tend to lose their focus and disorient from the team when they are occupied with other tasks.
4. One respondent claimed that behaviours inherited from traditional, competitive and hierarchical organisational structures do not work with agile methodologies. Hidden agendas of managers, departments or team members, preventing others’ success in parallel with internal competition, chasing prestige and titles demotivates the team and has negative effect on the context of their communication.
5. One respondent claimed that the habit of trying to solve problems over mails instead of face-to-face communication harms the relationship of team members.
6. One respondent argued that when there are people from different organisations in the team, their different organisational culture causes problems and can cause conflicts. She commented that in a project, people from two different organisations that have different working hours had problems in fixing the time for the daily stand-up meetings.
7. One respondent argued that performance reviews and performance based key-metrics cause problems in agile teams. When performance reviews are prepared from other team members’ comments about a member, conflicts and problems with trust appears in the team. Moreover, he claimed that key-metrics of the project success should be oriented towards team commitment rather than personal success.

Otherwise, team members try to beat each other with respect to metrics and this causes members not to trust each other.

Process

Question of "What do you think about agile development process's role in team performance?" was asked to participants.

1. One respondent argued that a badly designed process slows down the workflow. Changes made later on a project takes more energy and time to adjust than to design the optimal process beforehand.
2. In addition, one respondent noted that changing the process requires alteration of the data collected so far. Adjustment of that data to new process wastes time the team should spend on the project tasks.
3. Another respondent claimed that when the process is hard, people start to take shortcuts. This causes them to make mistakes and lose the track in the project progress.

Planning

Question of "What do you think about planning of the project's role in team performance?" was asked to participants.

1. It is commented that every member of the team including the product owner should be present at the meetings. Planning meetings are most effective and fruitful when all of the team members have the same level of knowledge about the project and contribute to the plan.
2. One respondent argued that the project team should decide on the details of the project plan themselves. When externally prepared plans are imposed on teams, their sense of ownership over the project decreases.
3. One respondent said that agile projects are complex and unpredictable environments. Therefore, strict and unrealistic deadlines forced by the upper management to the team may cause trust problems between the two parties.
4. A respondent argued that agile process requires specific tasks in planning process and the team should be capable of handling those tasks. Team's ability to turn user stories into understandable and

implementable story points have direct effect on how the planning will be effective on the project.

Meetings

Question of “What do you think about planning of the project’s role in team performance?” was asked to participants.

1. A respondent claimed that only the people who has an interest from the meeting should attend. Not everyone should be invited to the meetings since uninterested people lose their focus and distract other people. Moreover, he claimed that people spend their time on meetings that they have nothing to do with while they can work on their own agendas.
2. A respondent argued that chatty people and people who do not obey the meeting rules distract others, causes meetings to extend and causes people to talk out of context topics.
3. One respondent said that meetings usually result in action items and especially after retrospective meetings, members should take action to improve themselves according to the feedbacks. Otherwise, purposes of the meetings cannot be met are and it’s benefits cannot be realised by the team.

Iterations

Question of “What do you think about iterations’ and their activities’ role in team performance?” was asked to participants.

1. Two respondents claimed that long iterations make people bored, causes the team to put a lot of story points in the iteration and makes it harder to adapt to changes and new requests during the iterations. Moreover, planning for long iterations are harder than shorter ones.
2. However, another respondent argued that short iterations cause the team to get stressed because of frequent deadlines.
3. One respondent claimed that when iteration lengths change frequently, teams experience adaptation problems. In contrast, another respondent claimed that iteration lengths should not be unchangeable since that will decrease the team’s ability to adapt to changing requirements.

Documentation

Question of “What do you think about documentation activities’ role in team performance?” was asked to participants.

1. One respondent argued that some of the documents are required to be too long and detailed. Those documentations are hard for the team to both read and write. Unless simple, information in the documents are not transferred as intended.
2. Another respondent claimed that sometimes organisation or project rules require documents that are unnecessary. Some of the documents are never read although the team members dedicate a good amount of time creating it. This causes the team to waste their time that they can spend on delivering their tasks.
3. One respondent claimed that some documents that are required by the team to complete their tasks arrive late, which causes the team to wait until it arrives.
4. Another respondent claimed that documentation process decreases the communication between the team members. Therefore, if the documentation load on the team is high, team is under the risk of losing their synchronization.
5. One respondent argued that software developers are not used to and do not feel motivated when making documentation. This causes them to do their work slower and lose their morale.

5 Discussion

5.1 Introduction

The purpose of this research was to combine literature knowledge and interviews with the professionals and derive the commonly conceived barriers against team performance in agile software projects. To this aid, an in-depth analysis of the scholarly articles were made for literature review and 8 different agile project managers from various industries were interviewed for data collection. This chapter elaborates upon the collected data and try to analyse it to form a conclusion. Three perspectives under which the barriers were identified will be discussed respectively.

Due to limitations of time and resources, making quantitative generalisations out of the collected data may not result in accurate and scientific results. Nevertheless, the diversity of industries, working culture, organisations and project types gave the author to compare and contrast respondents comments to arrive to some generalisations and conclusions that may give insight to future research.

5.2 Discussion

5.2.1 Interpersonal and Personal Perspective

Purpose of studying interpersonal and personal perspective was to identify the barriers forming as a result of interactions between the team members and team members' personal characteristics. Main topics discussed under this perspective were culture, support, leadership, communication, team ownership, learning and personalities. Comments were grouped and listed under those topics. Critical analysis and discussion of the collected data is below.

Culture

Not all of the interviewed project managers work (or have worked) with a team including people from different cultural backgrounds. Therefore it is hard to make a generalisation out of the collected data. However, from the people who works (or have worked) with a multi-cultural team perceives cultural differences not as a barrier in general. One of the barriers that was mentioned by several participants was the existence of hierarchical mind-set in team members. Moreover, some of the barriers that the participants related to culture can be considered as different personalities instead of cultural differences. Relating personal mismatches between team members to culture may be a wrong diagnosis in some cases and may result in the application of wrong action to prevent conflicts.

Support

Supporting behaviour is seen as one of the most important characteristic of agile teams. Maintaining the progression of the project to continue as planned, providing members with new knowledge and increasing the communication potential hence the creation of new ideas in the team are dependent on the level of support between the team members. However, the general response that was got from the interview participants is that team members are not supporting each other when it comes to giving feedback, help and ideas. Creation of an environment that encourages team members to spare their time, energy and knowledge to help each other is seen as a challenging task. There is no doubt that a suitable mind-set for this behaviour should be created and maintained throughout the project. However, from several reasons such as trying to keep up with the project agenda or being shy to ask for help supporting climate is seen as a challenging thing to maintain.

Leadership

Leadership concept in agile teams are defined differently from those of traditionally working teams. Agile teams require and demand less command and control over the team members. Giving the team members enough freedom to self-manage themselves instead of micromanagement from a higher position is a key to improve the creative environment in the team. Interviews showed that participants managed to overcome traditional mind-set in their approach against leading the team. According to participants, leadership is seen as an essential element in agile teams however their involvement with everyday tasks are limited to guidance, solving problems and protecting the team. Barriers they perceive with leadership are related to boss-type traditional management and how it demotivates people and prevents the team from being self-managing.

Communication

Communication was one of the most discussed topic in scientific literatures. Likewise, most of the respondents claimed that barriers with communication are the most important in all of the topics that they have discussed. They accept that communication is a key element in a creative and dynamic agile team. Barriers perceived under this topic are related to human behaviour and characteristic clashes between team members. They argue that creating the environment so that team members trust each other and feel confident to communicate is a challenging task. Conflicts are seen as one of the major problems against an effective communication environment.

Team Ownership / Self-Managing Behaviour

Self-managing teams are essential in agile culture. It is thought that when teams are given freedom to decide on their routines and actions to approach a goal, more creativity and enthusiasm can be obtained. Respondents appreciated the importance and benefits of having a self-managing team. However, they commented on the hardness of creating a self-managing mind-set that penetrates from team to individual level. Habits from traditional experiences or cultures makes it hard for project managers to sustain a self-managing environment.

Learning

Learning is essential in agile teams to sustain the knowledge level of every team member at the same level. Emphasis on constructive communication between the team members in the agile manifesto supports the high potential for learning in agile teams. Comments from the respondents revealed that learning process requires certain amount of effort to manage which means it does not always occur naturally in the teams. Therefore, unless the team is encouraged to learn and an environment for potential of learning is created, learning process would be not very effective. Moreover, information hiding from various reasons is seen as an important barrier.

Product Owner/Client

All of the respondents works (or have worked) as a scrum master. Moreover, some has experience of working as a product owner. Literature views product owners as a proxy between the customer and the team. Navigating the team to best suit the end-product to the customer's needs with setting goals and giving feedback on the progress are main responsibilities of the product owner. Most of the barriers from respondents related to product owner is that when the product owner starts to give directive and act as a project manager, team loses its motivation. Therefore, sometimes behaviour of the product owner against the team is seen as a barrier. Moreover, respondents expected a certain amount of training, knowledge and experience about managing the product, technologies used and directing the team.

Personal

Agile methodologies depend more on people rather than the guidelines. Therefore, characteristics of individuals and how those characteristics react with other individuals' play an important role in teams' performance. Respondents confirmed this commenting that personalities effect behaviour of members in the team and hence effect the team's performance. One widely discussed barrier here was people who are not open to be a team player. When those people prioritize their individual benefits over the team benefits, teams lose their focus and most of the respondents commented that one incompatible team member can bring the whole team down.

5.2.2 Materialistic Perspective

Purpose of studying materialistic perspective was to identify the barriers forming as a result of physical environment and usage of hardware & software tools. Critical analysis and discussion of the collected data is below.

Office environment and tools used in projects are important in every project since they can directly affect the speed and efficiency of the team members. Moreover, these factors influence the level of interpersonal interactions and their quality. Agile projects depend on collaboration and communication rather than the concentrated usage of standard tools. However, many project managers appreciate the help of right workspace arrangement and usage of planning tools to increase the quality of interactions, visualization and automation of planning and progress.

Interviews showed that the common expectation of project managers when it comes to material perspective is the arrangement of work environment. A common barrier that was identified is that team members not being co-located in an office, which in turn decreases their communication level. In addition, while hardware and related performance problems were not seen as a major barrier, ineffective or wrong usage of software tools is seen as a barrier that decreases the level of communication and correct perception of the project progress.

5.2.3 Agile and Organisational Process Perspective

Purpose of studying agile and organisational perspective was to identify the barriers forming as a result of application of the activities, preparation of artefacts and direct/indirect effects of organisational mind-set. Main topics discussed under this perspective were organisational culture, planning, meetings, iterations and documentation. Comments were grouped and listed under those topics. Critical analysis and discussion of the collected data is below.

Organisational Culture

Organisations have mind-sets that affect and gets affected by the people working under them. Therefore, every decision made and action taken are influenced by the cumulative mind-set formed in these organisations. Same project is likely to give different results when it is performed under another organisation even though same people work on it. Most of the interview participants stated that organisational culture the second most important factor after communication when it comes to team performance.

According to respondents, a common barrier related to organisational culture is organisation’s mind-set not being in line with the project teams’. Participants stated that when the organisation does not embrace the agile culture as a whole, decisions made and processes applied may be influenced by traditional methods. Upper management trying to micro-manage an agile team was one of the common examples. This inconsistency of thought between team level and organisational level causes teams to lose their efficiency in self-management and decision making.

Planning and Meetings

Agile process is different when it comes to planning activities. In addition to pre-project planning, iterative progress requires smaller scale planning activities at the beginning of each iteration. Therefore, agile planning process is different from traditional planning in terms of complexity and quantity. Moreover, agile methodologies introduce different types of meetings aiming to both increase the team's daily performance and help them manage their activities.

A general claim from respondents is that planning process' and meetings' complexity and frequency should fit the needs of teams. Therefore, unnecessary and requisite planning and meetings without the team's will are seen as a barrier. This perspective is to some level relevant to participants claims in organisational culture section. As with other decisions, project managers think that teams should decide on the agenda and context of planning and meetings.

Iterations

Iterative nature of agile methodologies allow teams to test and improve the prospective product many times before the release, allowing the team to better adapt to changes and requirements. Iteration length and number are decided according to the complexity of the project and available resources.

Comments from the respondents show that both long and short iteration lengths can be seen as barriers. It is observed that there is a trade-off between the possibility of motivation decrease in long iterations and increase in stress level in short iterations. Some of the respondents claimed that keeping the same level of progression speed in long iterations become challenging and some claimed that short iterations cause people to hurry and make mistakes as well as exhaust them. However, none of the respondents commented on any negative effect of large number of iterations. Therefore, iteration durations that are inconsistent with project complexity are seen as barriers.

Documentation

Documentation is mostly essential and inevitable parts of projects. It is seen as a way to store and transfer knowledge between people or entities. Moreover, standards like ISO and CMMI require specific documentation.

Most of the respondents from interviews see unnecessarily many and long documentation as a barrier that slows down the project progress and demotivates team members. A common comment was that they try to avoid documentation without causing loss of knowledge. Responses from organisational culture section are relevant with the comments made for the documentation. Sometimes, the documentation required by the standards of organisations cause teams to devote their effort that they can otherwise spend on the product. It was stated that there are many documents that are not used after they were written, yet they are written because it is a requirement.

5.2.4 General Discussion

Out of 8 respondents, 4 were from Turkey, 3 were from Sweden and 1 was from Finland. This gave the author to compare the collected data with respect to working cultures of the countries. There were different barriers stated under organisational culture, communication and leadership topics by respondents from different countries. Effects of hierarchical working culture in Turkey and flat working culture of Sweden and Finland can be seen when it comes to how upper management affects the project team, competition for supremacy in job title, self-management without requiring directives and being open and talking straight in communication. Agile methodologies do not offer different strategies for different working cultures. Therefore, it is likely that every culture adapts them according to themselves to get best out of it.

A common barrier that was stated is the difference between agile team's mind-set and organisational mind-set. Respondents claimed for several reasons related to the difference between traditional and agile decision making and management issues. Some of these issues were stated as micromanagement habit, being plan-driven instead of product-driven and giving too much importance to titles and hierarchy. This shows that, for principles of agile development to be applied in project teams, both the organisation and the team should embrace the agile culture. Migration of organisations to agile culture is a challenging task especially in organisations with a long history of waterfall mind-set.

Unlike project management guides like PMBoK and PRINCE2, agile methodologies do not have strict, complex and detailed rules for managing projects. Agility depends on adapting the product development process to constraints and resources available. Therefore, agility can be understood by the level project teams embrace the agile culture and principles instead of application of practices. This mind-set was also visible in the interviews. When respondents commented on the barriers about agile process, they claimed that planning activities, meetings, iterations and documentation should not be done according to pre-determined, unchangeable rules. Instead, they argued that the whole process can and should be adapted to best fit the project team's ease and increase end-product's quality. In other words, respondents were for keeping the process as simple and productive as possible avoiding the practices that provide less or no benefits.

To summarise, possible barriers stated in the literature review part are in line with the collected data in many angles. However, it is important to note that the number of samples of data is not enough to deduct quantitative generalisations.

6 Conclusion

6.1 Conclusion

This dissertation has investigated the barriers that prevents better team performance in agile software projects. First, a literature study approaching the research question from two angles was made. First angle was software project management and agile project management. Second angle was teamwork. Next, those two viewpoints were used to analyse literature on explicitly stated or possible teamwork barriers in agile software project management that have risk or causing teams to fail or give poor results. Aim of the literature study was to get an understanding of sources of possible barriers so that this research can criticize and add on the current knowledge base.

Barriers found in literature were grouped under groups and sub-groups that has logical relevance with project teams. Next, qualitative data was collected and examined to make a detailed study of those groups and sub-groups. Finally, collected data was critically analysed with respect to literature and ethnology of data sources. Findings were listed under logical groups without an effort to create a theoretical model.

It is important to note that observations and conclusions made in this work comes from data collected in interviews. Therefore, results are more suitable to evaluate as perceptions from players of the industry rather than statistical and universal facts. Moreover, this study is open to criticism that same results may not be achieved when the study is repeated.

To summarise findings of the research, agile software projects encounter different barriers throughout the project lifetime that directly or indirectly affect the project team and decrease their performance causing decrease in quality and stakeholder satisfaction, and increase in effort and time. A brief summary of the most discussed points grouped under three main groups are below:

Interaction based and personal barriers such as

- ineffective communication,
- unresolved conflicts,

- trust problems,
- inability to self-manage,
- inadequate leadership and product evaluation,
- inability to work in teams,

materialistic barriers such as

- wrong selection of software tools,
- unsuitable office environment,

organisational and process based barriers such as

- organisation mind-set being different from the agile team’s,
- less control of team over the process,
- unnecessary meetings and documentation,
- inadequate iteration lengths

This work contributes to existing knowledge of performance of agile projects by further studying the possible reasons for failing agile projects or agile projects with less end-product quality than expected.

6.2 Further Research

Due to limitations in time and resources, dissertation context was limited. Limitations are explained in Scope and Limitations chapter of methodology chapter. However, this research can be used as a step stone to study related contexts further. Author recommends further research of the research question using a larger sample set to arrive to a quantitative result and form a theoretical model. Moreover, research can be extended to cover more working cultures and industries to arrive to a more general and viable result. Finally, it is suggested that a research aiming to better understand the reasons behind those barriers and offer possible solutions should be conducted.

7 Bibliography

Acuña, S. T., Gómez, M., & Juristo, N. (2009). How do personality, team processes and task characteristics relate to job satisfaction and software quality? *Information and Software Technology, 51*(3), 627-639.

Ahmed, A. (2012). *Software Project Management: A Process-Driven Approach*: Auerbach Publishers, Incorporated.

Association for Project Management. (2006). *APM Body of Knowledge* (5. ed.): The Association.

Augustine, S. (2005). *Managing Agile Projects*: Prentice Hall.

Augustine, S., Payne, B., Sencindiver, F., & Woodcock, S. (2005). Agile project management: steering from the edges. *Commun. ACM, 48*(12), 85-89.

Blankenship, J., Bussa, M., & Millett, S. (2011). eXtreme Programming *Pro Agile .NET Development with Scrum* (pp. 29-51): Apress.

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *Software, IEEE, 22*(5), 30-39.

Brannick, M. T., Salas, E., & Prince, C. (1997). *Team Peformance Assessment and Measurement: Theories, Methods, and Applications*: Lawrence Erlbaum Associates, Incorporated.

Brooks, F. P. (1995). *The Mythical Man-month: Essays on Software Engineering*: Addison-Wesley Pub. Co.

BSA. (no date (Accessed: 12 April 2013)). Software Industry Facts and Figures, from http://www.bsa.org/country/Public\%20Policy/\~/media/Files/Policy/Security/General/sw_factsfigures.ashx

Burke, C., Stagl, K., Klein, C., Goodwin, G., Salas, E., & Halpin, S. (2006). What type of leadership behaviors are functional in teams? A meta-analysis. *The Leadership Quarterly, 17*(3), 288-307.

Cervone, F. (2011). Understanding agile project management methods using Scrum. *OCLC Systems & Services, 27*(1), 18-22.

Charette, R. N. (2005). Why software fails [software failure]. *Spectrum, IEEE, 42*(9), 42-49.

Chiang, I. R., & Mookerjee, V. S. (2004). Improving software team productivity. *Commun. ACM, 47*(5), 89-93.

Chin, G. (2004). *Agile project management: how to succeed in the face of changing project requirements*: Amacom Books.

Cohen, S. G., & Bailey, D. E. (1997). What Makes Teams Work: Group Effectiveness Research from the Shop Floor to the Executive Suite. *Journal of Management, 23*(3), 239-290.

Conboy, K., Coyle, S., Xiaofeng, W., & Pikkarainen, M. (2011). People over Process: Key Challenges in Agile Development. *Software, IEEE, 28*(4), 48-57.

Cooke, J. L. (2012). *Everything You Want to Know About Agile*: It Governance Limited.

Davis, A. M., Bersoff, E. H., & Comer, E. R. (1988). A strategy for comparing alternative software development life cycle models. *Software Engineering, IEEE Transactions on, 14*(10), 1453-1461.

de Sousa, A. F. M., & Almeida, A. M. P. (2011). Agile Management of Multimedia Projects: a Case Study. *International Journal of Information Processing and Management, 2*(3), 100-114.

Deeprose, D. (2001). *Making Teams Works*: AMACOM.

Derby, E., & Larsen, D. (2006). *Agile retrospectives: making good teams great*: Pragmatic Programmers, LLC.

DiCicco-Bloom, B., & Crabtree, B. F. (2006). The qualitative research interview. *Medical Education, 40*(4), 314-321.

Fellows, R. F., & Liu, A. M. M. (2009). *Research Methods for Construction*: Wiley.

Fernandez, D. J., & Fernandez, J. D. (2008). Agile Project Management - Agilism Versus Traditional Approaches. *The Journal of Computer Information Systems, 49*(2), 10-17.

Fitsilis, P. (2008). Comparing PMBOK and Agile Project Management software development processes. In T. Sobh (Ed.), *Advances in Computer and Information Sciences and Engineering* (pp. 378-383): Springer Netherlands.

Ganesh, N., & Thangasamy, S. (2012). Lessons Learned in Transforming from Traditional to Agile Development. *Journal of Computer Science, 8*(3), 389-392.

Gulliksen Stray, V., Moe, N., & Dingsøyr, T. (2011). Challenges to Teamwork: A Multiple Case Study of Two Agile Teams. In A. Sillitti, O. Hazzan, E. Bache & X. Albaladejo (Eds.), *Agile Processes in Software Engineering and Extreme Programming* (Vol. 77, pp. 146-161): Springer Berlin Heidelberg.

Highsmith, J. (2005). Decleration of Independence, from http://www.pmdoi.org/

Highsmith, J. (2009). *Agile Project Management: Creating Innovative Products*: Pearson Education.

http://agilemanifesto.org/. (2001). Agile Manifesto, from http://agilemanifesto.org/

Katzenbach, J. R., & Smith, D. K. (2005). The Discipline of Teams. (cover story). [Article]. *Harvard Business Review, 83*(7/8), 162-171.

Khazanchi, D., & Zigurs, I. (2006). *Patterns of Effective Management of Virtual Projects: An Exploratory Study*: Project Management Institute.

Koch, A. (2004). *Agile Software Development : Evaluating the Methods for Your Organization*. Norwood, MA, USA: Artech House.

Kraut, R. E., & Streeter, L. A. (1995). Coordination in software development. *Commun. ACM, 38*(3), 69-81.

Levesque, L. L., Wilson, J. M., & Wholey, D. R. (2001). Cognitive Divergence and Shared Mental Models in Software Development Project Teams. *Journal of Organizational Behavior, 22*(2), 135-144.

Locke, K. D. (2001). *Grounded theory in management research*. London [u.a.]: SAGE Publ.

Marks, M. A., Mathieu, J. E., & Zaccaro, S. J. (2001). A Temporally Based Framework and Taxonomy of Team Processes. *The Academy of Management Review, 26*(3), 356-376.

Millett, S., Blankenship, J., & Bussa, M. (2011). *Pro Agile .NET Development with SCRUM*: Apress.

Moe, N. B., Dingsøyr, T., & Dybå, T. (2009). Overcoming Barriers to Self-Management in Software Teams. *Software, IEEE, 26*(6), 20-26.

Moe, N. B., Dingsøyr, T., & Dybå, T. (2010). A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology, 52*(5), 480-491.

Moe, N. B., Dingsøyr, T., & Røyrvik, E. (2009). Putting Agile Teamwork to the Test – An Preliminary Instrument for Empirically Assessing and Improving Agile Software Development. In P. Abrahamsson, M. Marchesi & F. Maurer (Eds.), *Agile Processes in Software Engineering and Extreme Programming* (Vol. 31, pp. 114-123): Springer Berlin Heidelberg.

Nandyal, R. S. (2003). *People Capability Maturity Model*: Tata McGraw-Hill.

O'Connor, M. K., Netting, F. E., & Thomas, M. L. (2008). Grounded Theory: Managing the Challenge for Those Facing Institutional Review Board Oversight. *Qualitative Inquiry, 14*(1), 28-45.

Payne, G., & Payne, J. (2004). *Key concepts in social research*: SAGE Publications.

Pichler, R., & Schulze, S. (2005). Book Reviews: Agile Project Management: Creating Innovative Products by Jim Highsmith, and Agile Project Management with Scrum by Ken Schwaber. *Journal of Product Innovation Management, 22*(4), 371-373.

Rico, D. F., Sayani, H. H., Sutherland, J. V., & Sone, S. (2009). *The Business Value of Agile Software Methods: Maximizing ROI With Just-in-Time Processes and Documentation*: J. Ross Publishing, Incorporated.

Rubin, K. S. (2012). *Essential Scrum*: Addison Wesley Professional.

Rudestam, K. E., & Newton, R. R. (2007). *Surviving your dissertation: a comprehensive guide to content and process*: SAGE Publications.

Saunders, M. N. K., Lewis, P., & Thornhill, A. (2009). *Research methods for business students*: Financial Times/Prentice Hall.

Schwaber, K. (2009). *Agile Project Management with Scrum*: Microsoft Press.

Schwaber, K., & Sutherland, J. (2012). *Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, And Leave Competitors In the Dust*: Wiley.

Stober, T., & Hansmann, U. (2010). *Agile Software Development: Best Practices for Large Software Development Projects*: Springer Berlin Heidelberg.

Strauss, A., & Corbin, J. M. (1998). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*: SAGE Publications.

Takeuchi, H., & Nonaka, I. (1986). The new new product development game. [Article]. *Harvard Business Review, 64*(1), 137-146.

Truex, D. P., Baskerville, R., & Klein, H. (1999). Growing systems in emergent organizations. *Commun. ACM, 42*(8), 117-123.

Vlaanderen, K., Jansen, S., Brinkkemper, S., & Jaspers, E. (2011). The agile requirements refinery: Applying SCRUM principles to software product management. *Information and Software Technology, 53*(1), 58-70.

Wysocki, R. K. (2010). *Effective Software Project Management*: Wiley.

Young, S. M., Edwards, H. M., McDonald, S., & Thompson, J. B. (2005). Personality characteristics in an XP team: a repertory grid study. *SIGSOFT Softw. Eng. Notes, 30*(4), 1-7.

8 Appendix

8.1 Appendix A – Interview Topics

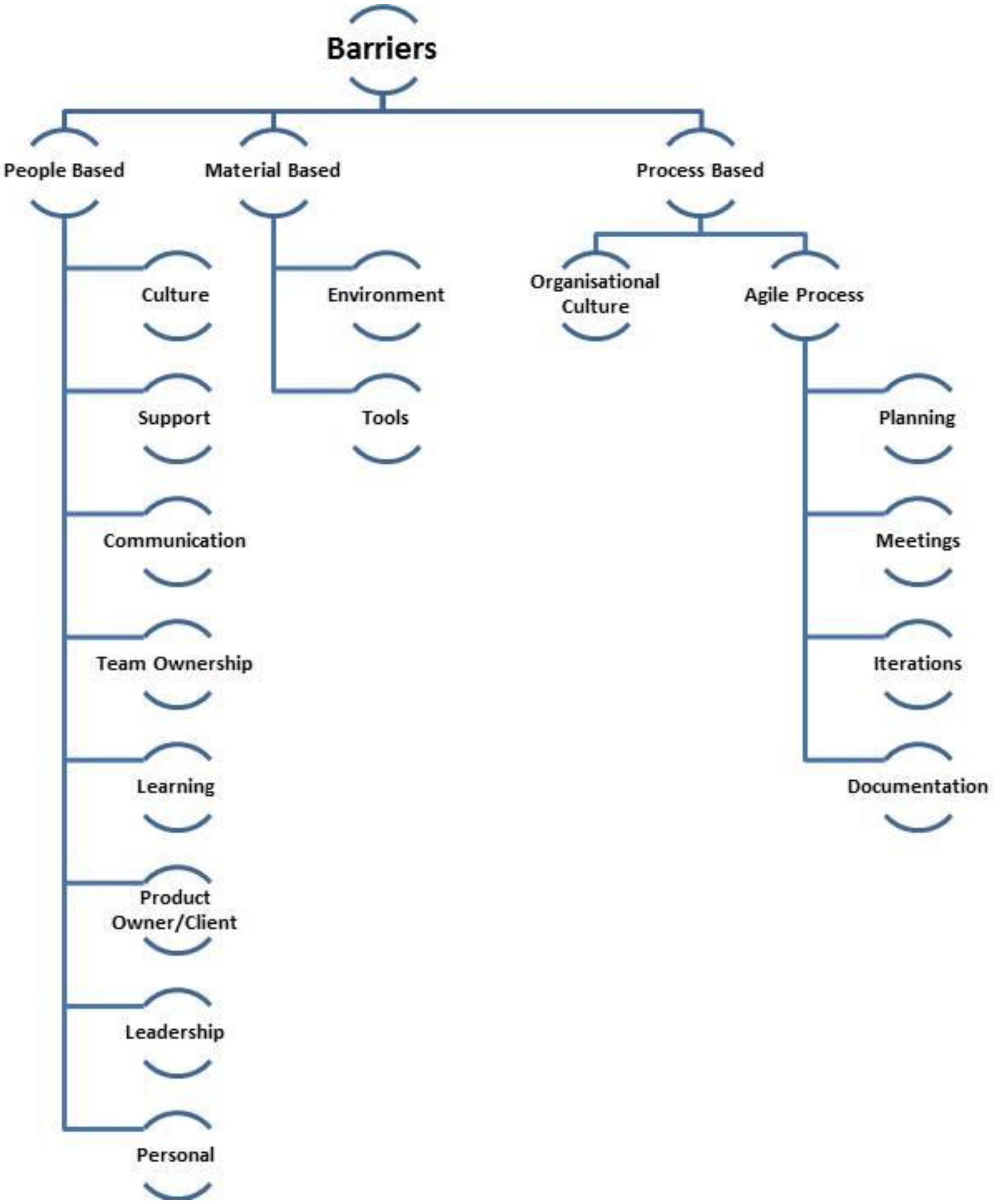
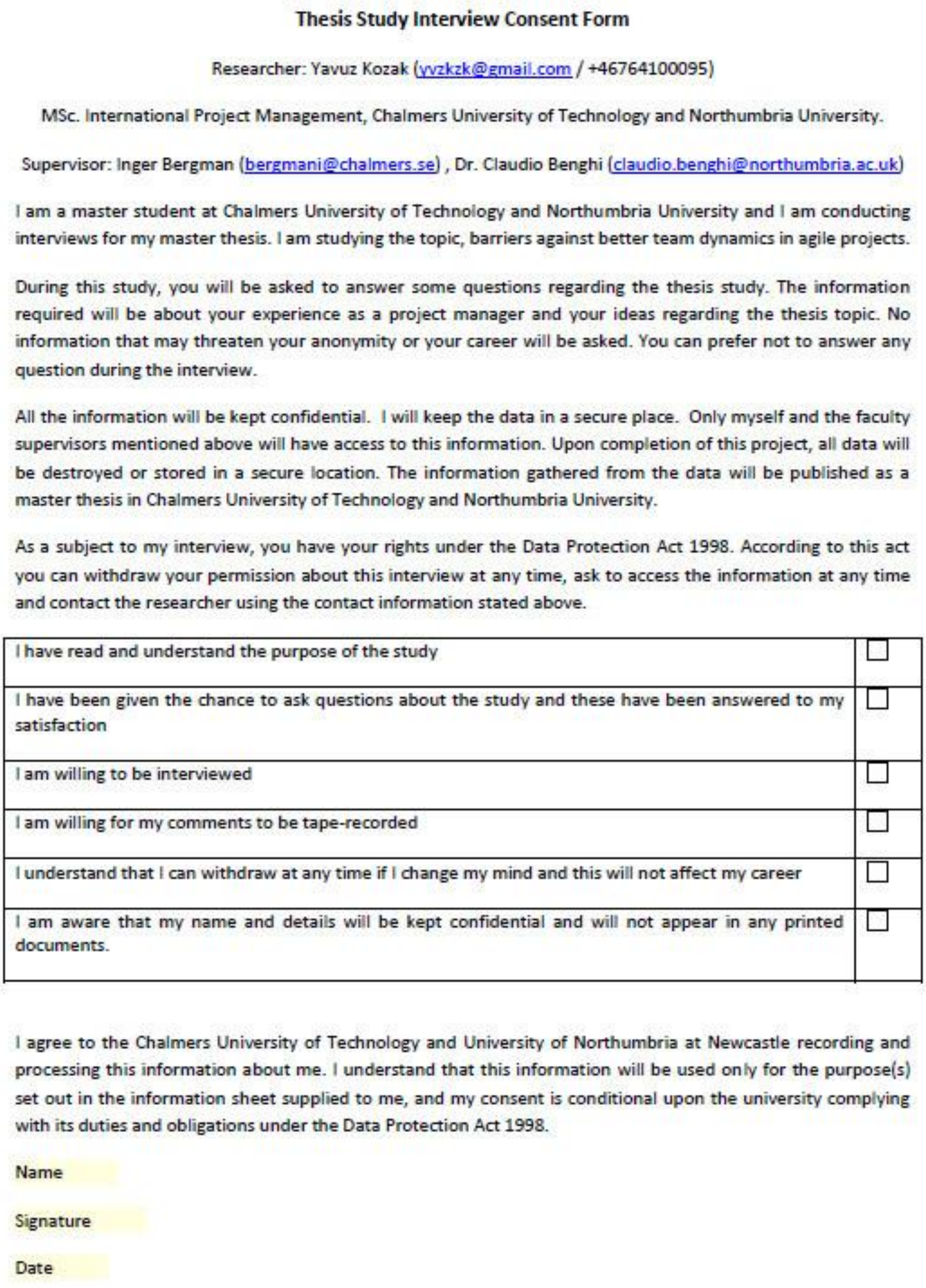


Figure 4 – Interview Topics

8.2 Appendix B – Interview Consent Form



Thesis Study Interview Consent Form

Researcher: Yavuz Kozak (yvzkzk@gmail.com / +46764100095)

MSc. International Project Management, Chalmers University of Technology and Northumbria University.

Supervisor: Inger Bergman (bergmani@chalmers.se) , Dr. Claudio Benghi (claudio.benghi@northumbria.ac.uk)

I am a master student at Chalmers University of Technology and Northumbria University and I am conducting interviews for my master thesis. I am studying the topic, barriers against better team dynamics in agile projects.

During this study, you will be asked to answer some questions regarding the thesis study. The information required will be about your experience as a project manager and your ideas regarding the thesis topic. No information that may threaten your anonymity or your career will be asked. You can prefer not to answer any question during the interview.

All the information will be kept confidential. I will keep the data in a secure place. Only myself and the faculty supervisors mentioned above will have access to this information. Upon completion of this project, all data will be destroyed or stored in a secure location. The information gathered from the data will be published as a master thesis in Chalmers University of Technology and Northumbria University.

As a subject to my interview, you have your rights under the Data Protection Act 1998. According to this act you can withdraw your permission about this interview at any time, ask to access the information at any time and contact the researcher using the contact information stated above.

I have read and understand the purpose of the study	□
I have been given the chance to ask questions about the study and these have been answered to my satisfaction	□
I am willing to be interviewed	□
I am willing for my comments to be tape-recorded	□
I understand that I can withdraw at any time if I change my mind and this will not affect my career	□
I am aware that my name and details will be kept confidential and will not appear in any printed documents.	□

I agree to the Chalmers University of Technology and University of Northumbria at Newcastle recording and processing this information about me. I understand that this information will be used only for the purpose(s) set out in the information sheet supplied to me, and my consent is conditional upon the university complying with its duties and obligations under the Data Protection Act 1998.

Name

Signature

Date

8.3 Appendix C – Agile Manifesto and 12 Principles

Agile Manifesto and 12 principles behind the manifesto can be found in http://www.agilemanifesto.org web site.

8.3.1 Manifesto for Agile Software Development

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.

8.3.2 Principles Behind the Agile Manifesto

We follow these principles:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.