

Susanna Broman

Agile Pitfalls

Helsinki Metropolia University of Applied Sciences

Master of Business and Administration

Business Informatics

Thesis

21.5.2017

Author Title	Susanna Broman Agile Pitfalls
Number of Pages Date	51 pages 21 May 2017
Degree	Master of Business and Administration
Degree Programme	Business Informatics
Instructor	James Collins, Senior Lecturer
<p>The purpose of this study was to identify the most common pitfalls of agile software development and to provide a checklist for overcoming these issues. The use of agile methods has been a rising trend in the software development and the number of agile pitfalls organizations are facing is endless, but there are a lot of same mistakes many organizations are doing one after another. There is no case company involved in the study but the subject was chosen due to authors own interest in agile methods.</p> <p>Qualitative research methodology was used in this study. The research data consisted of interview discussions with five agile professionals representing different organizations.</p> <p>The results of the interviews revealed the most common issues organisations are facing in agile software development. The interviewees had rather similar views and it became obvious that the same issues were taking place repeatedly in different organizations. The interviewees embraced agile in many ways but felt that it was often used without careful consideration. In addition, a lack of sufficient pre-requisites and knowledge was experienced, leading to issues with quality, communication and efficiency.</p> <p>The author recommends that organizations planning to go agile would use a checklist to ensure awareness of the possible pitfalls and the way they can be avoided. It is recommended to consider whether it is reasonable to use agile instead of traditional methods, what kind of agile approach to select and to create a change management strategy with an execution plan.</p>	
Keywords	Agile software development, agile pitfalls, change management.

Contents

1	Introduction	1
1.1	Background	1
1.2	Business challenge	2
1.3	Objective	2
1.4	Output	2
1.4.1	Context	3
1.4.2	How the thesis progresses	3
2	Method and material	4
2.1	Research design	4
2.2	Data collection and analysis	5
3	Preliminary literature	8
3.1	Agile weaknesses	8
3.1.1	Organizational and management related challenges	8
3.1.2	People related challenges	9
3.1.3	Process related challenges	9
3.2	Preliminary literature review conclusions	10
4	Current state analysis	11
4.1	Agile strengths	11
4.2	Agile weaknesses	12
4.3	Conclusions	14
5	Conceptual framework	16
5.1	Software development life cycle	16
5.2	Agile software development	17
5.2.1	The agile manifesto	18
5.2.2	Scrum	19
5.3	Traditional software development	21
5.3.1	Waterfall model	21
5.4	Differences between agile and traditional software development	22
5.5	Change management	23
5.5.1	Change management strategy	24
5.6	Transforming to agile	26

5.6.1	Conclusion	28
6	Proposal building	33
6.1	Initial proposal	33
6.1.1	Selection of a method	34
6.1.2	Selection of an approach	35
6.1.3	Creation of a change management strategy	35
6.1.4	Creating and following the execution plan	35
7	Proposal validation	36
8	Final proposal	37
8.1	Selection of a method	44
8.2	Selection of an approach	45
8.3	Creating a change management strategy	45
8.4	Creating and following the execution plan	46
9	Discussion and conclusions	47
9.1	The credibility of the study	47
9.2	Conclusions	48
	References	49

1 Introduction

In today's world, organizations in different branches are using more and more agile ways of working. As the operational environment is constantly changing and organizations are forced to keep up the pace to stay alive, they might not be able to survive by following only the old inflexible methods. However, thorough consideration and preparation needs to be done before changing into agile. In many cases, organizations are so used to follow traditional models, such as waterfall, that they do not realize that the organization itself needs to be changed as well, not just the method they are following. The number of agile pitfalls organizations are facing is endless, but there are a lot of same mistakes many organizations are doing one after another. These common issues are the most interesting ones and therefore highlighted in this thesis.

In this thesis, the most common pitfalls of agile software development are investigated and suggestions how to avoid them are introduced. The thesis is not related to any specific organization or technology but common issues identified by having some informal interview discussions. First, a preliminary literature was written in order to have a hunch on common issues, before starting interview discussions and preparing current state analysis. Based on current state analysis conclusion, topics for the literature review were identified. After literature review, initial proposal for tackling the most common agile pitfalls in advance was prepared and validated by agile professionals. These agile professionals were partly representing same persons that were interviewed for the current state analysis. Finally, after initial proposal was validated, the final proposal was written.

1.1 Background

The topic for the thesis was decided based on author's own passion and interest. The author has been working as a scrum master and wanted to gain more knowledge in order to develop the use of agile methods in her own job. She had experienced a lot of positive implications because of agile way of working instead of traditional methods. However, she had faced also some severe issues and wanted to drill down to learn whether other people are having same experience and how these could be avoided.

This thesis is not built around any case organization and therefore people interviewed are representing couple of different organizations. Interviewed people were chosen based on suitable background and their willingness to participate and they are all having agile experience. Though the thesis is not done to any specific organization, the outcome of it can be considered as a checklist for any person or organization that are either planning to go agile or already are using agile but facing issues and would like to improve way of working.

1.2 Business challenge

The business challenge of this thesis is that managers in software development adopt agile as some sort of cure-all without consideration to the challenges that are likely to be encountered for this particular field of work. The business challenge is not related to a single organization but common issues.

1.3 Objective

The objective of the thesis is to develop a checklist, how to overcome issues in agile software development. Target audience for the checklist are people like the author; individuals who are using agile methods in their job and would like to improve the way of working to embrace agile benefits. However, the checklist could be useful also to persons and organizations that are only planning to go agile.

1.4 Output

The output of the thesis is a validated proposal in a form of a checklist, answering to a question how to overcome some of the most common issues in agile software development. By taking the checklist into a consideration when planning to go agile, organizations can avoid the most common agile pitfalls.

1.4.1 Context

As the use of agile methods has been a rising trend in many organizations in all branches and not least in the software development, agile pitfalls is very actual topic. Despite the popularity of agile, surprisingly many organizations do not familiarize themselves with careful preparations but are getting an illusion that agile simply means lightening or even skipping the planning and project management tasks. Software development is demanding and there are many possible stumbling blocks that are not fading away by just saying that traditional methods will be replaced with agile. Agile methods are not curing all the problems and not leading to a successful end without seriously going into it. The output of this thesis should help organizations to understand the pre-conditions of agile and things to consider before going agile software development.

1.4.2 How the thesis progresses

In the next chapters, first the research method and material used is explained. Then, the summary of the preliminary literature is written, following by the current state analysis. After and based on the current state analysis, the conceptual framework is introduced. Last, an initial proposal and its validation is described ending to a final proposal in addition to conclusions.

2 Method and material

This chapter describes the research design and data collection methods.

2.1 Research design

Qualitative research method is used due to its suitability to the thesis. In addition to the current state analysis and literature review, also preliminary literature review is done to gain a hunch of the current issues. The design of the research process is illustrated in below figure.

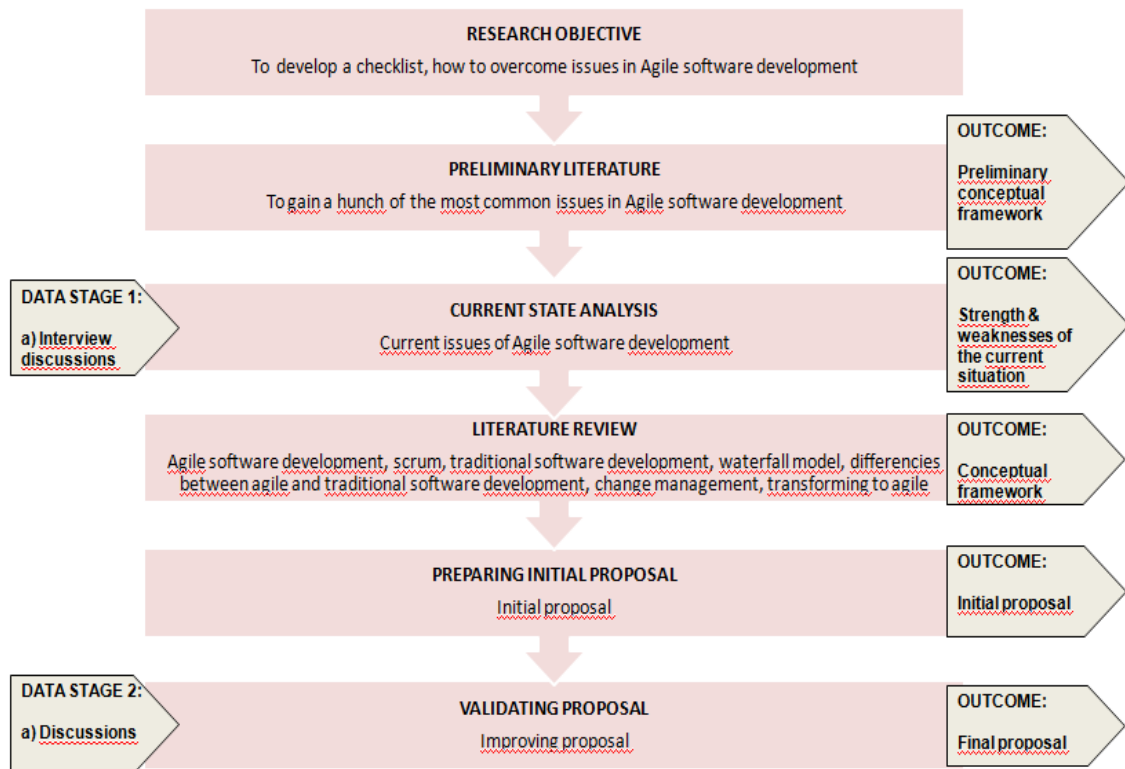


Figure 1. Research design of the thesis.

First preliminary literature review is carried out in order to get a hunch of the most common issues in agile software development. Though the issues that are collected from the literature are not exactly similar to the ones identified based on interview discussions, they are still directional and a good starting point. In the literature, issues are introduced

from all over the world, from different kind of organizations and different technologies. Most of all, the issues in the literature are mainly more generic compared to the ones identified by discussions with individuals.

After the preliminary literature review, the current state analysis is drawn up based on informal interview discussions with people involved in agile software development. Current state analysis is introducing the current strengths and weaknesses of agile software development. Interviewed people are representing scrum masters and developers from different organizations.

In the next phase of the thesis, a literature review is done; the main concepts related to the summary of the current state analysis are explained, such as agile software development, scrum, traditional software development, waterfall method, differences between agile and waterfall, change management and agile transformation. The literature review is targeting to conceptual framework that will be a base for the initial proposal, a checklist how to overcome most common issues in agile software development. Initial proposal is validated by couple of the interviewed persons; the initial proposal is fine-tuned based on their comments and the outcome is the final proposal.

When considering the validity of the research process it can be stated that above mentioned was valid for this case because there was no case company involved. Also, the subject is so new and broad that discussions instead of a questionnaire were more suitable.

2.2 Data collection and analysis

Data collection for data stage 1 was done via informal face-to-face discussions with people involved in agile software development. With some of the people, discussions were not just one-time but continued couple of times. Originally the purpose was to have few more discussions, but it became obvious rather soon that the answers were started to repeat themselves. Hence it did not make sense to continue discussions. There were total five people discussed with, representing both scrum masters and developers. As illustrated in below picture, four scrum masters and a developer were interviewed, from

couple of different organizations. Discussions were done informally and incognito in order to get honest and independent opinions from people. Field notes were done by the author to record the discussions.

Data was analysed by picking-up the main points from the answers and to coming back to those in cases where it was not clear enough what the interviewee was trying to say. All the interviewees were having their own point of view, a very unique way to express things and hence it required some analysis and re-discussions to be able to crystallize the main points.

After the main points from the answers were picked-up, they were categorized under few topics to be able to identify the areas of issues. This was helping to understand the big picture and the areas where the biggest issues were lying. Also, the identification of the literature topics was much easier after the categorization.

Data stage	Who	How	How data was recorded	Outcome
Data stage 1	Scrum Master 1	Informal discussions	Field notes	Current state analysis, strengths and weaknesses of agile software development
	Scrum Master 2	Informal discussions	Field notes	
	Scrum Master 3	Informal discussions	Field notes	
	Scrum Master 4	Informal discussions	Field notes	
	Developer/Scrum Team member	Informal discussions	Field notes	

Table 1. Data stage 1, informal interview discussions.

As shown in below table, data stage 2 was done by introducing the thesis as a whole and especially the initial proposal to two of the interviewees participating to data stage 1. Informal discussions with two individuals were done and the author prepared field

notes. Their comments and suggestions were taken into account when the final proposal was prepared. Comments and suggestions were compared to the theory of the thesis and the initial proposal to figure out how they could be put into practice and fine-tune the initial proposal.

Data stage	Who	How	How data was recorded	Outcome
Data stage 2	Developer/Scrum Team member	E-mail evaluation	Written/e-mail	Final proposal
	Scrum Master 2	Informal discussion	Field notes	

Table 2. Data stage 2, validation of initial proposal.

3 Preliminary literature

In this chapter, findings from the preliminary literature are introduced. The purpose of this chapter is to gain preliminary information before starting the interviews and current state analysis, to have a hunch of the most common agile issues.

3.1 Agile weaknesses

In the study of Gandomani, Ghani, Ziaei and Zulzalil, (2013), the obstacles and issues in agile software development are categorized under four themes; organizational and management related challenges, people related challenges, process related challenges and technology and tools related challenges. Many of the current challenges are stem from the culture and structure of the organization which is serving needs of traditional methods.

3.1.1 Organizational and management related challenges

Organizational culture is affecting to agile transform. Organizational culture is a vague term covering numerous things such as prevailing attitudes, norms and values (Iivari & Iivari 2010).

Gandomani, T. et al. (2013) are using a term “The agile transformation process” when discussing about organizations moving from traditional methodologies into agile. Organizations are often making a mistake by underestimating the difficulty of the agile transformation process and not investing it; this is making challenges even more difficult.

Organizational issues in agile software development are coming from too narrow thinking of the meaning of agility. Organizations are often stating they are agile though it usually means only software development. The software development is failing in agility in cases where the organization around it is not agile enough. The software development projects and teams cannot fully use their agile potential unless the organization is not supporting them and getting rid of traditional thinking and old habits. When the agile software development team is lacking agile support from their organization, it tends to

lead situations where people are not feeling safe to share identified issues and mistakes; this is reducing agility and impacting to end results (Gothelf, J. 2014).

According to Moczar (2013), agile is promising too much when stating that it would be a solution to problems faced with traditional methods; Moczar (2013) has identified several times that agile is partly falling to same issues than with other methods. Organizations are counting too much on pure agile method and forgetting the importance of agile thinking. In cases where only the agile method has been followed without changing the mindset as well, it has sometimes led even to bigger catastrophes than by using traditional methods and changed the good intentions totally upside down. One of the common issues is that organizations are not considering carefully whether the use of agile is worthwhile (Moczar, L. 2013).

3.1.2 People related challenges

Since agile is all about people, people related challenges are playing a significant role especially in cases where the organizations have earlier been using traditional software development methods. One of the common people related weaknesses is the difficulty for people to change their mindset and behaviour into agile mode. During agile transformation, there is not always enough training and coaching from agile expertise though it would be needed. People related issues are concerning both customers and vendors and both can have overwhelming impacts (Gandomani, T. et al. 2013).

3.1.3 Process related challenges

For instance, the agile principle of early and continuous delivery is sometimes leading to too hasty outcome in detriment of quality. This principle is allowing developers to neglect to bugs. The consequence of too fast delivery might be the growth of defect backlog, ending up to excessive work (Moczar, L. 2013).

The manifesto for agile software development is encouraging to “development over planning”. This has been often an issue though the original idea has been to make things easier. There are often issues because the size of the changes is varying from a tiny to huge ones. Though agile is welcoming changes even late in the development, it is still

commonly causing problems because the development is constantly ongoing and there might be unsolved defects making it even harder to success in agile (Moczar, L. 2013).

The plan to have a totally self-organized team without a project manager who would be responsible for the whole project is not always working as desired. What happens often is that the scrum master is forced to act as a project manager to keep things going on, but without a project manager mandate. For instance, the prioritization of the tasks to be done is an issue faced in the real world; often time-pressure is so high that an additional prioritization is needed. In practise, it is difficult for developers to manage all the priorities and dependencies by themselves (Moczar, L. 2013).

3.2 Preliminary literature review conclusions

The outcome of the preliminary literature review are some the most common weaknesses of the agile software development on a high-level. The weaknesses of agile software development are for instance:

- organizations are not agile enough and therefore not able to provide support for the agile software development teams
- people with experience on traditional software development are not able to get rid of their old habits and mindsets and preventing the successful use of agile
- agile processes are not properly used due to lack of agile knowledge

When reading the results of the preliminary literature review, it needs to keep in mind that though the issues mentioned are partly similar than in the current state analysis, they cannot totally match due to fact that CSA is done by interviewing Finnish IT-professionals while literature is from the wider perspective. Still, the preliminary literature is providing a hunch, a useful overview.

4 Current state analysis

In this chapter, the most common strengths and weaknesses of agile software development are being introduced. The current state analysis is prepared based on informal and anonymous interview discussions.

4.1 Agile strengths

Based on interview discussions, the following strengths of agile software development were identified; intense and good cooperation, easiness to plan work in small pieces, possibility to correct mistakes rather easily and quickly, allocated resources, if preconditions are in place the quality is usually good. Though above mentioned are considered as strengths, they still cannot be taken for granted but can be achieved only by treating agile method with conscious. Agile strengths can turn to weaknesses in a quick manner if agile principles are not followed actively.

First, people discussed with were having positive experience on cooperation and communication between different parties such as the project team and customers. Especially when sitting at the same premises and having extended face-to-face communication, the cooperation has been much more informal and therefore better compared to traditional approaches. Communication can be done without delays and so called Chinese whispers –effect can often be avoided, also threshold to open discussion is low. One of the scrum masters highlighted the easiness of the cooperation when all project members are sitting on the same premises; he had experienced that good cooperation usually requires people locating on same premises and as soon as part of the scrum team is located for instance in another country, communication gets poor. All interviewees mentioned good cooperation and communication as the most valuable thing agile can offer. However, they all had experienced the fragility of good cooperation, meaning it can easily be spoiled. This will be elaborated more in the next subchapter.

Another identified strength of agile software development is the easiness to plan work in small pieces. This is a great advantage because the changes in the schedule and error estimates are not causing as much issues as with traditional methods. The so-called snowball effect can be avoided rather easily and the possibilities to adjust the overall schedule works better. One of the scrum masters stated that it is unrealistically to even

think that all the smallest details could be planned in the beginning of the project due to nature of the software development and especially regarding bigger software projects. Hence, he appreciated the possibility that agile is offering: to plan work in pieces.

Third strength of the agile software development was identified to be the good possibilities to correct mistakes and bugs easily and relatively early. People were having unpleasant experience on traditional methods where mistakes are not often noticed until at the end of the project, but they considered agile way of working to enable faster issue fixing. People noticed that for example in scrumming, you are learning sprint by sprint and eventually be a master. The scrum master 1 was praising agile due to its mercifulness; in he's experience, software development done by traditional methods is harsh and punishing people for all mistakes they are doing especially in the beginning of the project, when agile method is often offering a possibility to fix mistakes during the coming sprints. He's opinion was that in agile software development; the learning curve of the scrum team members is much better because it is actually possible to learn by mistakes fast within the project and not only after the project is about to end or even finished.

Allocated resources are also one of the agile strengths people mentioned. Allocated resources are a great benefit because they know the product that is developed but also other project members, enabling to proceed smoothly. In perfect situations resources are allocated 100% to the agile project itself, this is something that is unfortunately not always happening but when it does, it makes agile life easy. One of the interviewees, a scrum master, stated that everything is much easier by using agile because there are designated resources and they are mainly allocated to the same project.

4.2 Agile weaknesses

Despite all the strengths, there are also several weaknesses in agile software development, such as:

- agile methodology is used though there are not prerequisites
- lack of sufficient planning or documentation or testing
- too early delivery
- communication and cooperation issues due to resources located in different places
- issues due to cultural differences when projects are international

- resources not always able to concentrate 100% to agile work due to other responsibilities
- changes in staffing affecting agile projects heavier than traditional ones
- agile methodology and principles not known
- bigger risks to break existing functionalities because the big picture not always known due to constant changes done

Three of the most common weaknesses are explained in detail in this chapter, though there is not much difference between the answers by the interviewees. Also, to mention, some of the weaknesses are almost overlapping.

One and the most common of the weaknesses observed and discussed was that in many cases, all agile resources are not 100% allocated to agile work due to other responsibilities. This is causing delays to the development work and makes it difficult to plan schedules. Even one person with less than full-time allocation may cause tremendous issues. As the developer that was interviewed said, since things are unfortunately often depending on individuals, the non-attendance of even one person can spoil the whole thing and undercut the benefits of agile.

Even too early delivery, meaning lack of sufficient planning, documentation and testing is also a big issue regarding agile software development. Some of the people interviewed stated this issue to be concerning the whole project, covering all the steps and starting from the project planning; they felt that in some cases, the project team thought that the use of agile would justify defective quality. Though agile is encouraging to iterations and welcoming changes over planning, this was sometimes misused. When using agile, there is sometimes pressure to deliver outcomes earlier than what would be wise and realistic, leading to careless development and lack of proper testing. Especially lack of planning and documentation is sometimes making bug fixing difficult and causing too much dependency on individuals. Without proper planning, there are often conflicts between the development done by other people within the same agile team or even other projects. Poor planning is often leading to quality issues and bugs as well. In cases where also the documentation is negligible, the defect fixing is even more painful and time consuming. In addition, the software around is constantly changing, making it harder to identify the root cause for issues and corrective actions.

The third biggest weakness discussed was the use of agile methodology without having preconditions to adopt it. People were having bad experience of projects originally planned to be done with traditional methods but for varied reasons the method was changed to agile; these situations were often leading to confused situation where agile method was supposed to be followed but the organization around the project group was not acting agile at all. Some of the people were considering agile as a trendy concept that is rather often used without really focusing on it and the conditions it is requiring. Typically, the thought is to run a project like with waterfall method but without any specifications and with minimal testing. One of the scrum masters was even having experience on agile team developers not at all familiar with the agile method itself, leading to waist of valuable time reserved for the development work. He used a lot of time during several sprints for teaching agile principles and scrumming to other team members.

4.3 Conclusions

Strengths and weaknesses based on interview discussions are listed in below table.

Strengths	Weaknesses
Intense and good cooperation	Agile used though not prerequisites
Easier to plan work in small pieces (development items, sprints)	
Possibility to fix mistakes rather quickly	Too early delivery, lack of sufficient planning/documentation/testing
Allocated resources	In case resources are located in different places, communication and cooperation becomes harder
In case preconditions are in place, quality is usually good	In case resources are located in different countries, cultural differences are causing issues
Possibility to learn fast by mistakes	All resources are not able to concentrate 100% to agile work due to other responsibilities
	Changes in staffing is affecting agile projects more heavily than traditional projects

	Agile methodology and principles are not known well enough
	Bigger risks to break existing functionalities
	The big picture is not always known due to constant changes done

Table 3. Strengths and weaknesses of agile software development.

Interviewees were overall satisfied with the quality of work in agile projects. They all though in many cases, agile approach works better than traditional one. Due to designated resources and emphasizing the communication and cooperation, risk to fail is less. Especially good and intense cooperation and designated resources were appreciated. However, there are several weaknesses as well, such as all resources may not be 100% allocated to agile work due to other responsibilities, misusing agile approach by working carelessly and using agile though all the preparation work was not done. As the interviewees were speculating, most of the issues are due to lack of proper preparations and underestimation of agile approach. Interesting observation was that people identified more agile issues than successes.

An interesting observation is that many of the strengths and weaknesses are opposite to each other, meaning that the advantages of agile can be gained only with careful consideration and preparation, and without this they can turn into weaknesses. When rushing to agile without preconditions in place, the results are not always positive as expected. When discussing with people about what should be done differently to succeed with agile, a common denominator seems to be that better change management and learning agile deeper would be needed. In the next chapter, literature review based on findings from the current state analysis is introduced.

5 Conceptual framework

In this chapter, a conceptual framework of the thesis is being introduced. Topics are identified based on conclusions of the current state analysis. The purpose of this chapter is to support the understanding of the thesis and to prepare the proposal.

The current state analysis revealed that the most common issues are related, on a high-level, to either agile transformation, the differences between agile and traditional methods or change management.

5.1 Software development life cycle

Software development consists of the following stages:

1. Requirements and analysis
 - a. Decision on what the software should do
 - b. Clarifying the needed input- and output data
2. Design
 - a. Breaking down the details
 - b. Decision on desired layout
 - c. Planning the programming part
3. Implementation
 - a. Implementing the program code
4. Testing
 - a. Multiple testing scenarios
5. Evolution and maintenance:
 - a. Corrective
 - b. Perfective
 - c. Adaptive

(BBC Bitesize 2017).

5.2 Agile software development

The idea of the agile software development is to have an adaptive team which can deliver frequently and rapidly and welcome changes in the requirements. The advantages of the agile software development are “the ability to respond to the changing requirements of the project” (Balaji, S. & Murugaiyan, S. 2012) and the improved communication between the customer and the development team. Agile method is usually more profitable and suitable for smaller projects. One of the issues in agile software development is the demand for senior-level resources; agile developers should be able to do decisions and be self-imposed (Balaji, S. & Murugaiyan, S. 2012).

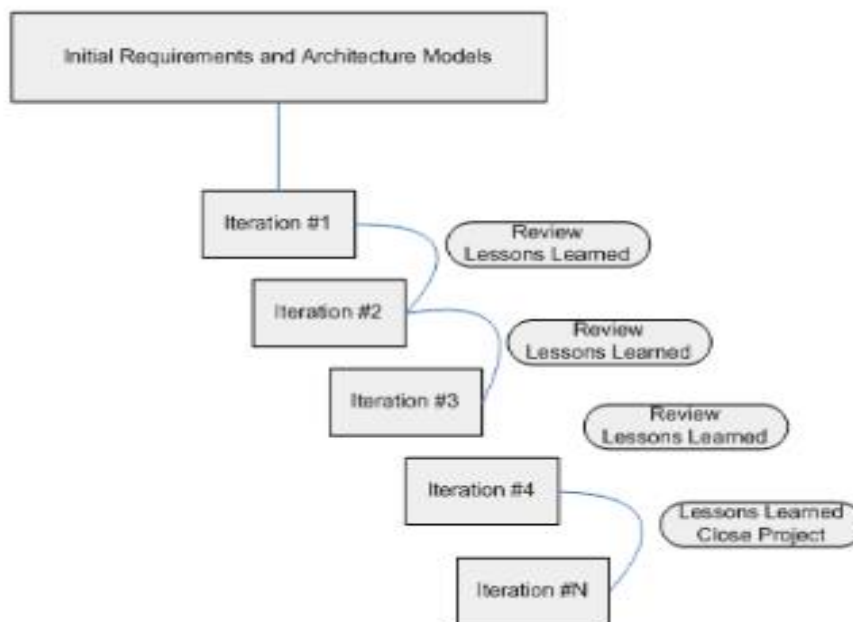


Figure 2. Agile model life cycle (Balaji, S. & Murugaiyan, S. 2012).

5.2.1 The agile manifesto

Manifesto for agile software development:

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

(Agilemanifesto.org 2001).



Figure 3. The agile manifesto (Lichtenberger, A. 2014).

12 Principles behind the agile manifesto:

12 Principles behind the agile manifesto

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

Figure 4. 12 Principles behind the agile manifesto (Agile alliance 2016).

5.2.2 Scrum

Scrum was founded in 1990s by Ken Schwaber and Jeff Sutherland and it is the most popular agile methodology worldwide. It is used mostly in software development and information technology but also for example in product development (Denning, S. 2015).

According to the official scrum guide,

“Scrum (n): A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.” (Schwaber, K. & Sutherland, J. 2013).

Scrum has empirical and iterative approach, aiming to control risks and highlight predictability. According to empirical approach, there are three main principles to follow; adaptation, inspection and transparency. The purpose of transparency is to keep the whole process visible to the people who are either performing or accepting the work. Inspections are referring to the idea that scrum artifacts should be inspected enough to detect the unwanted side effects but not exaggerate. Adaptation is aiming to adjustment of the artifact in case the inspection is revealing that the artifact is unacceptable (Schwaber, K. & Sutherland, J. 2013).

The product owner, development team and a scrum master are formulating a self-organizing scrum team that should not be depending on outsiders. The scrum teams are having needed competencies to deliver the artifacts incrementally and iteratively. Continuous feedback is desired to develop the competence and productivity (Schwaber, K. & Sutherland, J. 2013).

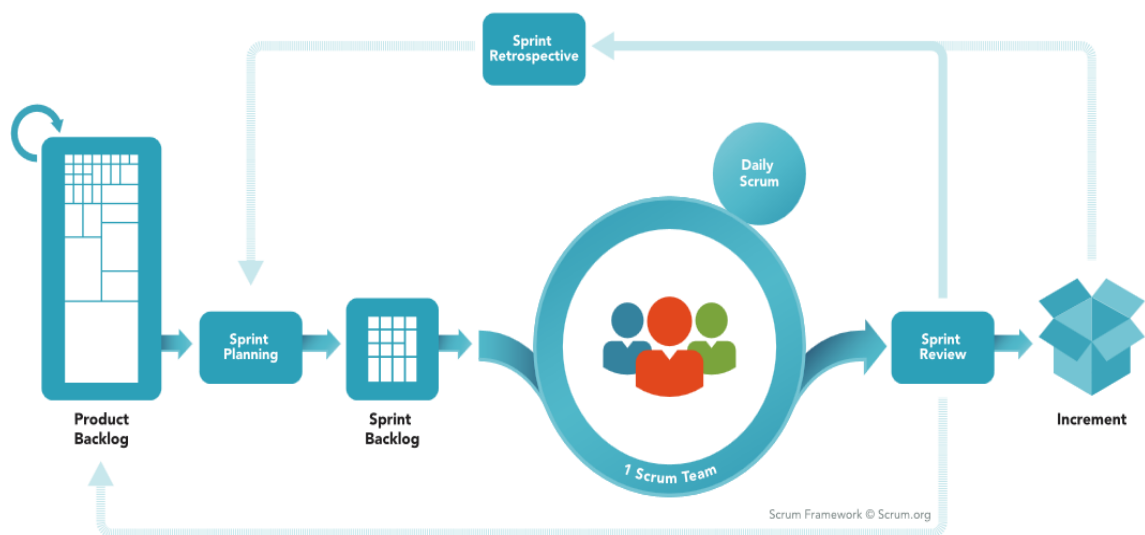


Figure 5. Scrum framework (Scrum.org 2016).

5.3 Traditional software development

Traditional software development is approaching things from the predictive point of view. Traditional software development is based on detailed plan with a complete list of items that must be developed. All the changes are going through a change control management (Ghilic-Micu, B. et al. 2013).

5.3.1 Waterfall model

Traditional and one of the oldest and most popular ways of software development is the document driven, sequential waterfall method. The catch of the waterfall method is to follow the pre-defined stages and milestones and to invest on early planning. An output of a stage is an input for the for the coming stage. At first, requirements are gathered and right after that follows the design phase. After the design, the implementation i.e. coding and testing is done and the final phase is handing to maintenance (Bhuvanewari et al., 2013; Balaji, S. & Murugaiyan, S. 2012).

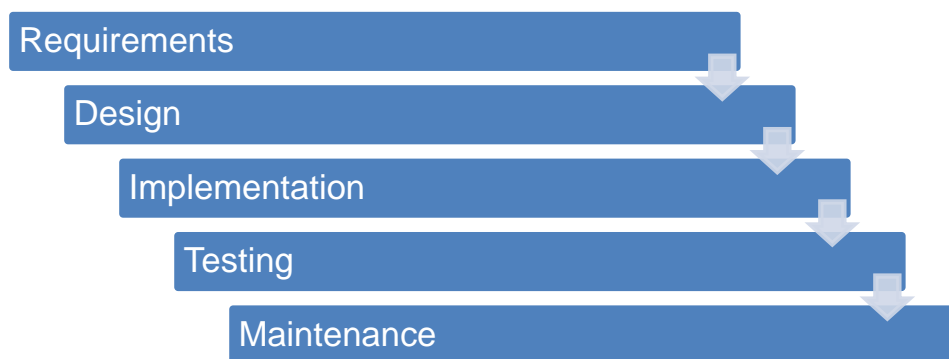


Figure 6. The waterfall model.

The advantage of the waterfall method is the easiness to understand and implement it due to its linear model. Waterfall is useful on mature products and weaker teams can benefit more from it. However, one centric pain point of the waterfall method is the unrealistic expectation that requirements in the beginning of the project could be strict and unchangeable, leading to issues in the latter phases of the projects. In this model, issues cannot usually be solved in one phase completely, leading to quality issues in the final

outcome. As the final deliverable, i.e. the actual software is delivered at the end of the project, possible issues are identified late leading to expensive changes (Bhuvaneshwari et al., 2013; Balaji, S. & Murugaiyan, S. 2012).

5.4 Differences between agile and traditional software development

Project Component	Traditional	Agile
Control	Process centric	People centric
Management Style	Command-and control	Leadership-and-collaboration
Knowledge Management	Explicit	Tacit
Role Assignment	Individual – favours specialisation	Self-organising teams – encourages role interchangeability
Communication	Formal and only when necessary	Informal and continuous
Customer’s Involvement	Important, usually only at the analysis phase of the project	Critical and continuous
Project Cycle	Guided by tasks or activities	Guided by product features
Development Model	Life cycle model (Waterfall, Spiral, or some variation)	The evolutionary-delivery model
Desired Organisational Form/Structure	Mechanistic (bureaucratic with high formalisation)	Organic (flexible and participative encouraging cooperative social action)
Technology	No restriction	Favours object-oriented technology
Team Location	Predominately distributed	Predominantly collocated
Team Size	Often greater than 10	Usually less than 10
Continuous Learning	Not frequently encouraged	Embraced
Management Culture	Command and Control	Responsive
Team Participation	Not compulsory	Necessary
Project Planning	Up-front	Continuous
Feedback Mechanisms	Not easily obtainable	Usually numerous available
Documentation	Substantial	Minimal

Table 4. Differences between agile and traditional software development (Conboy, K. et al.).

5.5 Change management

According to Kotter, change management “refers to a set of basic tools or structures intended to keep any change effort under control. The goal is often to minimize the distractions and impacts of the change.” (2011).



Figure 7. Kotter's 8-step process for leading change (Kotter international).

CHANGE MANAGEMENT AS A PRE-PLANNED PROCESS

TYPICAL MISTAKE: JUMPING FROM (1) TO (7)

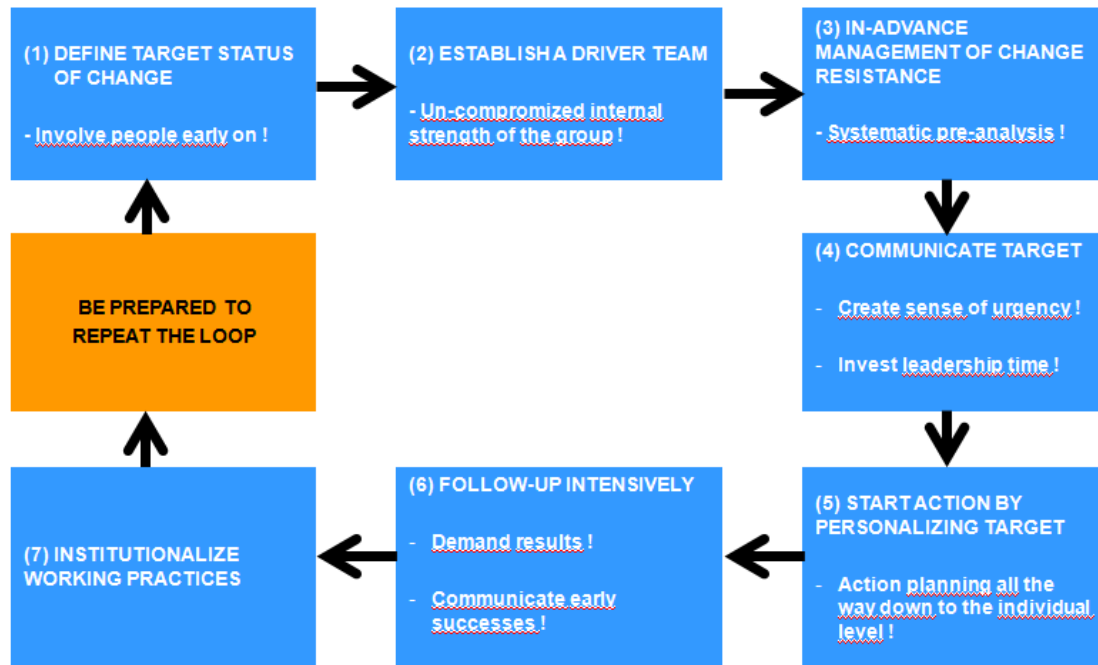


Figure 8. Change management process (Rohweder 2016).

5.5.1 Change management strategy

There are several alternative approaches to change and the selection should be done case by case, taking all the circumstances into account. Lockitt (2014) has roughly divided change management strategies into five different approaches; directive, expert, negotiated, educative and participative. However, these strategies are not exclusive and can be used alongside. One of the change management tasks is to make a decision what strategy or strategies to use and how and when to implement them (Lockitt, B. 2014).

One of the five strategy approaches, directive strategy emphasizes the authority of the managers, even without other people involved in the decision making. This approach is allowing fast change but not taking other involved people's opinions into account. The disadvantage of this strategy is often strong change resistance and lack of ideas from other stakeholders (Lockitt, B. 2014).

Another strategy approach, expert, is looking the change management from the problem solving point of view and it is suitable especially for the technical cases such as new systems being introduced. There are often specialists leading this kind of changes which is bringing both advantage and issues as well; though this approach is enabling rather quick implementation, affected people may not share same views than experts driving the change (Lockitt, B. 2014).

Negotiating strategy approach is highlighting the negotiating between the management and people affected. The management is letting stakeholders to express their views and is willing to do compromises regarding how and what is to be done. By following this approach, the change is having slower tempo and the predictability of the outcome is not complete, however people affected are more involved and there is less change resistance (Lockitt, B. 2014).

Educative strategy is trying to change people's way of thinking, leading them to support the change. Different kind of activities is used within this strategy, such as training and sweet talking by experts and consultants. Naturally, this approach is time-consuming but as an advantage, it is involving and committing people and reducing the amount of change resistance (Lockitt, B. 2014).

In participative strategy, all affected people are involved and their opinions are taken into account. In case experts and consultants from the outside are used to facilitate the change management process, they are not allowed to do any decisions. This approach is offering a possibility to learn and grow up, for both individuals and the organization around them. In addition, it is committing people and making them to support the change. As a disadvantage, this kind of change process is taking a lot of time and may be expensive (Lockitt, B. 2014).

Strategy	Advantages	Disadvantages
DIRECTIVE	Relatively fast	Ignores the views of those affected by change
EXPERT	Use relevant expertise. Small groups required. Relatively fast to implement.	Expertise may be challenged. Resistance of those not consulted.
NEGOTIATED	Change recipients have some say. Resistance to change likely to be reduced (or areas of disagreement highlighted)	May be relatively slow. Anticipated change may have to be modified.
EDUCATIVE	People committed to change.	Relatively slow. Likely to require more resources and more costs involved.
PARTICIPATIVE	Change more likely to be accepted. More people committed to change. More opportunities for individual and organisational learning.	Relatively slow to implement. More complex to manage. Will require more resources. Increased costs.

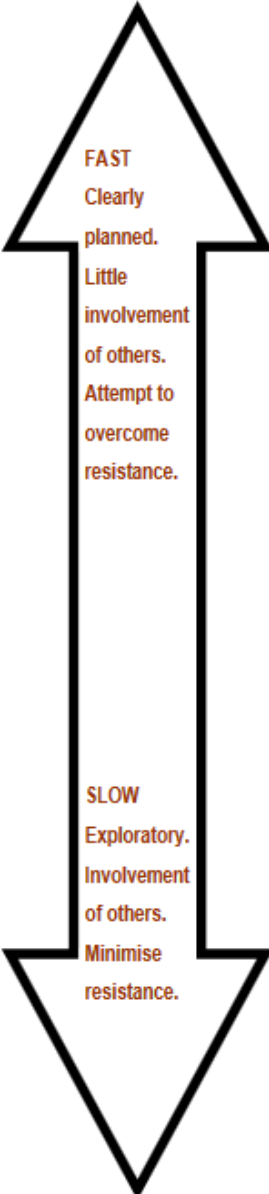


Figure 9. Overview of the five change management strategies (Lockitt, B. 2014).

5.6 Transforming to agile

When moving to agile, a strategy for the agile change management is needed. Agile transformation is socio-technical process that requires a lot of time and patient. There are three different approaches to use when moving to agile; tailoring, localization and adoption. Tailoring is aiming to fewer changes in the organization and it was popular especially in the days when agile methods were introduced. Tailoring approach may not

always be the best way to implement agile but rather a way to have the disciplined process and agile side by side (Gandomani, T. et al. 2012).

Instead of tailoring, localization is accepting essential changes but not all agile activities. Some parts of agile might be ignored totally and some are customized. Especially in organizations that are taking their first steps towards agile and lacking experience, some practices are still done by following traditional ways (Gandomani, T. et al. 2012).

Adoption approach is emphasizing major changes to adapt organizations with agile. When using adoption approach, agile methods are tried to be used completely without any limitations. Agile adoption is considered as the best way to achieve agile method (Gandomani, T. et al. 2012).

Challenges in agile transformation have been categorized as follows: management and organizational challenges, people challenges, process challenges and technology related challenges. Impacting to people's mindset is one of the biggest challenges; it is impossible to achieve overnight and besides time, it requires mentoring as well. Individuals as members of a project team may cause severe issues because of their habits, ambitions and different cultural backgrounds. Coaching towards agile is unique comparing to other methodologies and therefore requires an experienced and professional mentor in order to succeed. When changing to agile, people must change and forget old habits and roles; for example, project managers with strong experience in traditional methods must learn new way of working and forget being a commander. Also, the role of a customer is changing radically because of the agile way of working, forcing them to contribute in a different way (Gandomani, T. et al. 2012).

From the management point of view, tacit knowledge and minimal documentation are causing issues and can be treated as barriers. Still, one of the biggest management relates agile issues to be considered is the group decision making which is totally opposite when comparing to the traditional software development. Besides group decision, also letting individual project team members do self-governing decisions is part of agile but can sometimes be hard for the management to implement in practice (Gandomani, T. et al. 2012).

In many organizations, changing processes from traditional life cycle model to more iterative and evolutionary agile is difficult. This change affects many levels such as strategies, people's roles and measurement practices. In organizations where operations are spread to different locations, process related barriers towards agile transformation are playing even a bigger role and challenges regarding communication and cultural differences needs to be taken into account as well (Gandomani, T. et al. 2012).

5.6.1 Conclusion

As a conclusion, transforming from the traditional software development methods to agile is never easy but a time-consuming process that needs to be treated with a conscious and understand the importance of it. Everybody involved in agile transformation needs to be aware of the challenges and sufficient training and coaching must be provided. In addition, as there are several different agile methods to choose, organizations should carefully study them to find the most suitable one for them. All in all, in order to succeed, agile transformation requires a professional change management strategy, plan and resources.

Change management strategy from a wider perspective is mandatory for successful agile transformation. Purely technical point of view, concentrating on software development process is not sufficient but all aspects as illustrated in below picture should be taken into account. Agile transition is change oriented, not methodology oriented process that is touching all levels in the organization (Gandomani, T. et al. 2012).

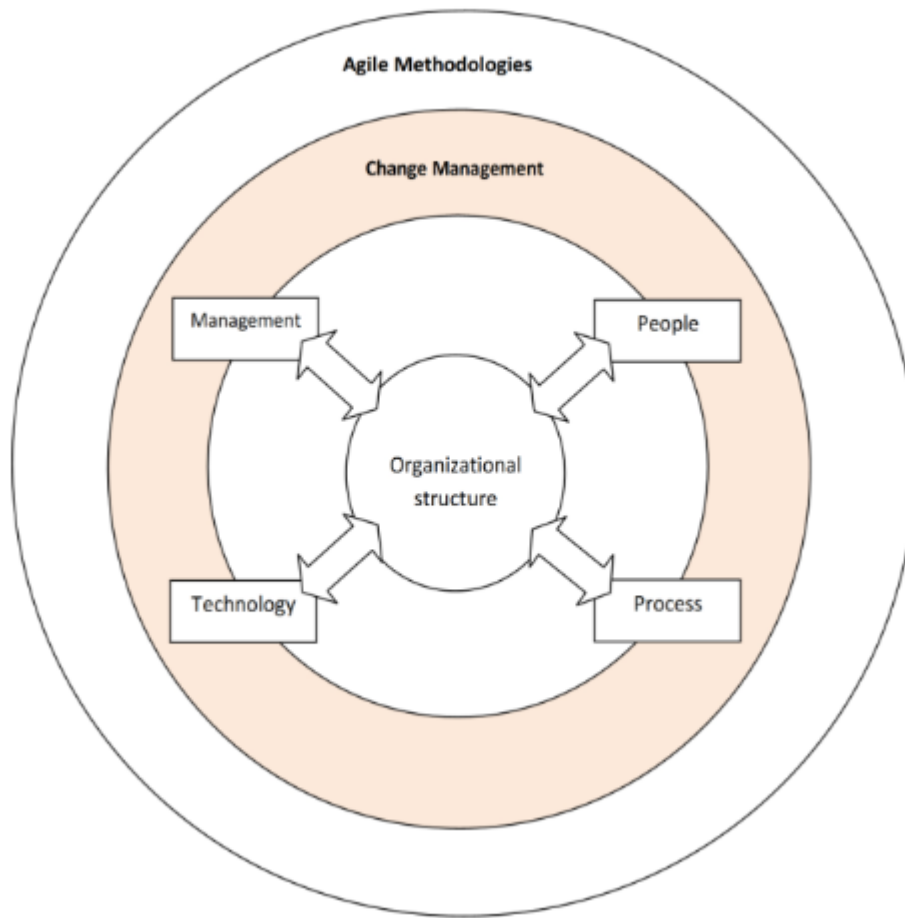


Figure 10. General plan of change management strategy (Gandomani, T. et al. 2012).

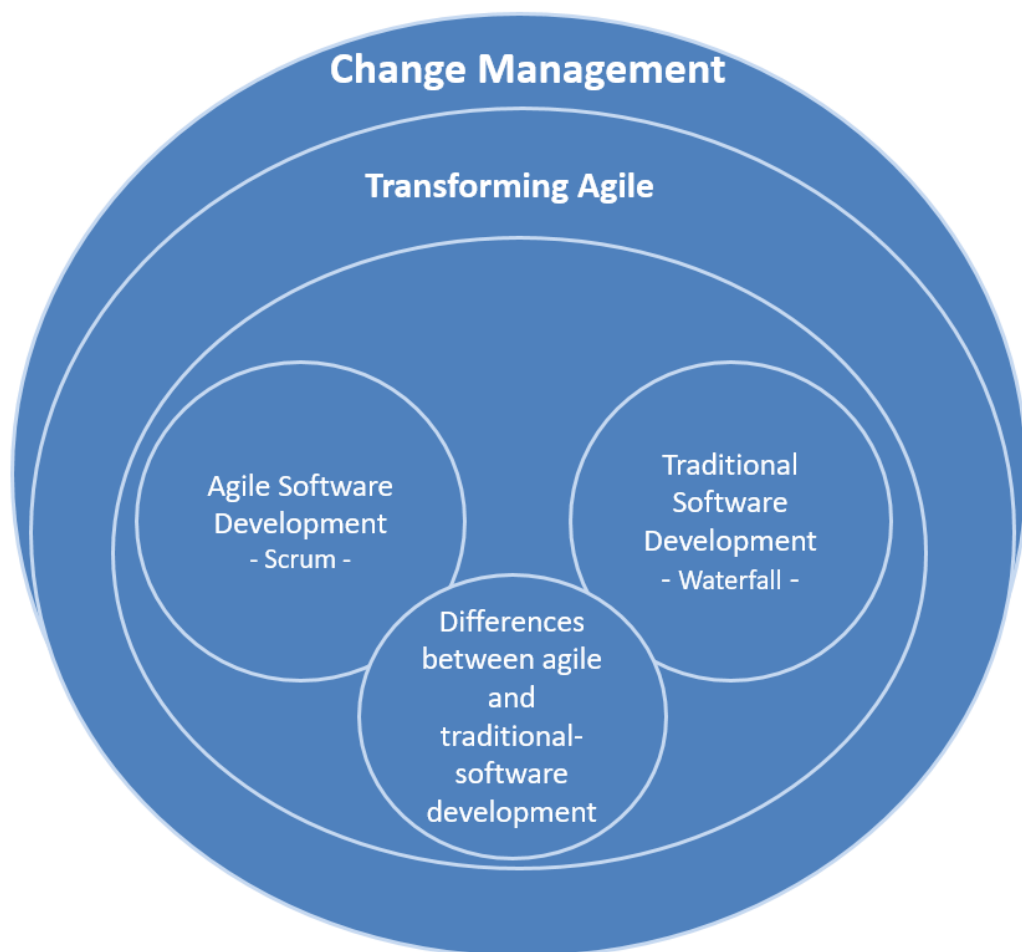


Figure 11. Theory of the thesis.

Weaknesses	Corresponding theory	Corresponding phase in the proposal
Too early delivery, lack of sufficient planning/documentation/testing	Agile software development & transforming agile & change management	Selection of a method & selection of an approach & creating a change management strategy & creating and following the execution plan
In case resources are located in different places, communication and cooperation becomes harder	Agile software development & transforming agile & change management	Selection of a method & selection of an approach & creating a change management strategy &

		creating and following the execution plan
In case resources are located in different countries, cultural differences are causing issues	Agile software development & transforming agile & change management	Selection of a method & selection of an approach & creating a change management strategy & creating and following the execution plan
All resources are not able to concentrate 100% to agile work due to other responsibilities	Agile software development & transforming agile & change management	Selection of a method & selection of an approach & creating a change management strategy & creating and following the execution plan
Changes in staffing is affecting heavily to agile projects than traditional	Agile software development & transforming agile & change management	Selection of a method & selection of an approach & creating a change management strategy & creating and following the execution plan
Agile methodology and principles are not known	Agile software development & transforming agile & change management	Selection of a method & selection of an approach & creating a change management strategy & creating and following the execution plan
Bigger risks to break existing functionalities	Agile software development & transforming agile & change management	Selection of a method & selection of an approach & creating a change management strategy & creating and following the execution plan
The big picture not always known due to constant changes done	Agile software development & transforming agile & change management	Selection of a method & selection of an approach & creating a change

		management strategy & creating and following the execution plan
Agile used though not pre-requisites	Agile software development & transforming agile & change management	Selection of a method & selection of an approach & creating a change management strategy & creating and following the execution plan

Table 5. CSA vs theory vs proposal.

6 Proposal building

In this chapter, initial proposal to overcome issues in agile software development is introduced. Initial proposal is prepared based on data 1 which is current state analysis and literature review.

6.1 Initial proposal

Initial proposal is trying to take all the previously introduced aspects in to account to offer a useful checklist. Initial proposal is telling who, what and when certain actions needs to be done. The aspect “why” is not mentioned in below figure because the lack of the case company; the thesis is based on common issues and not related to a specific organization.

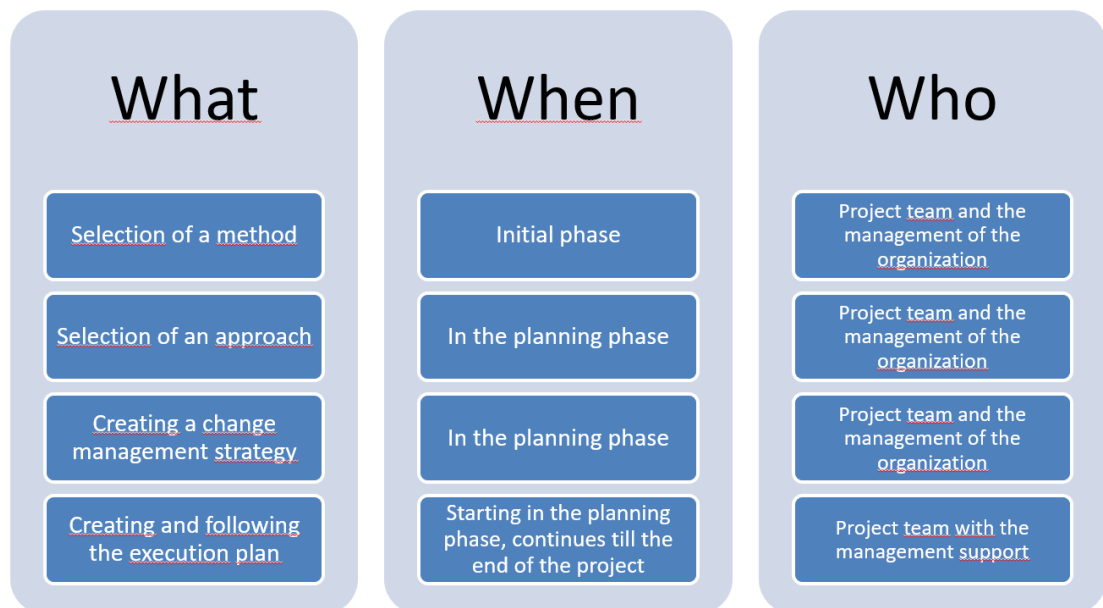


Figure 12. Initial proposal.



Figure 13. Initial proposal.

6.1.1 Selection of a method

There are several things that organizations and individuals should be taken into account when planning to go agile. At first, a careful consideration which one, traditional or agile method would be preferable, should be done. Comparison between these two different methods should always be done case by case and understand the unique features in every project. There are cases where agile is not suitable at all despite of all the benefits it is offering. When doing the comparison, also the characteristics of the organization are crucial; some organizations are more traditional and rigid, having a lot of bureaucracy. It can be extremely challenging or even impossible to bring agility to organizations like this.

6.1.2 Selection of an approach

After careful consideration and selection of the method, desired approach should be defined. As introduced in earlier in the literature review, there are roughly three alternatives to select from; tailoring, localization and adoption. When selecting the approach, all aspects must be considered realistically, from the project and organizational point of view. One major thing impacting to the selection of the approach is the former experience on agile or the lack of it.

6.1.3 Creation of a change management strategy

A change management strategy should be created by considering all known and common challenges, meaning management-, organizational-, people-, process and technology related aspects should be considered. The creation of a change management strategy must be done in the planning phase, after the method to follow and the approach has been chosen, before the actual project starts. As explained in the literature review, first the most suitable change management strategy approach to achieve the desired change needs to be defined. When defining the strategy, all aspects of the change must be taken into consideration; the organizational culture, the scale of the change, expected change resistance, schedule, budget and risks of the change.

6.1.4 Creating and following the execution plan

An execution plan is needed, together with the active follow-up. It is crucial to plan in detail how the actions will be executed; the plan itself is not enough but it needs to be followed-up as well.

7 Proposal validation

The initial proposal is validated and commented by two of the interviewees participating in current state analysis; a developer and a scrum master 2. Validation was done via e-mail and by having informal discussion. Also comments from the thesis supervisor was received.

The developer commented that the initial proposal was good and realistically. She is working in a software development industry and using agile methodology in her work currently. Her company is struggling with same issues mentioned in this thesis and hence planning to start implementing similar phase than the selection of approach -phase in the initial proposal; they came into a conclusion that a phase like this is a must in order to avoid facing same agile pitfalls over and over again. The company did the decision without knowing the initial proposal introduced in this thesis, which is a notable example of the necessity and usefulness of this kind of a checklist.

The developer was thinking that the way agile methodologies are used in Finland may be different than in other countries and especially other continentals. In her experience, Finnish companies are not yet too familiar with agile software development and therefore the initial proposal would probably not be as usable in other countries but suitable in Finland.

The scrum master 2 evaluated the initial proposal as simple and doable. In her experience, this kind of checklists needs to be simply enough and the correlation between commonly known issues and the checklist needs to be clear to get people interested about it. She stated that in case companies would not like to execute all phases, they could still pick-up certain phase or phases and execute them individually; this is an alternative that should be highlighted and explained.

The thesis supervisor highlighted the lack of the named resources; in the initial proposal, there is only mentioned either project team or management. However, this is not sufficient but leaves it too vague and raise a question "how to make sure things will be done". In addition, the thesis supervisor was missing a more concrete checklist with actions and their sub-tasks.

8 Final proposal

Since there was not identified any major changes during the proposal validation, the final proposal is rather like the initial proposal with a comment that in case companies do not want to implement all the phases, they can also pick-up an individual phase and execute it; it is not recommended but better than ignoring the whole checklist.

There is also more depth added to make sure that things will be done; there must be a responsible person pointed-out, regarding all the steps in the final proposal. In the initial proposal, instead of individuals, there were mentioned that either a project team or management should be responsible for certain steps. It was too vague definition creating a risk that things will not necessarily be done and certainly not on time. In the final proposal, it is suggested that named person can be either from the project team or management; it is depending on the project and organization which one is more preferably.

A detailed check-list with all sub-tasks is also added to the final proposal. The checklist is covering all stages of the proposal and its purpose is to offer more concreteness.

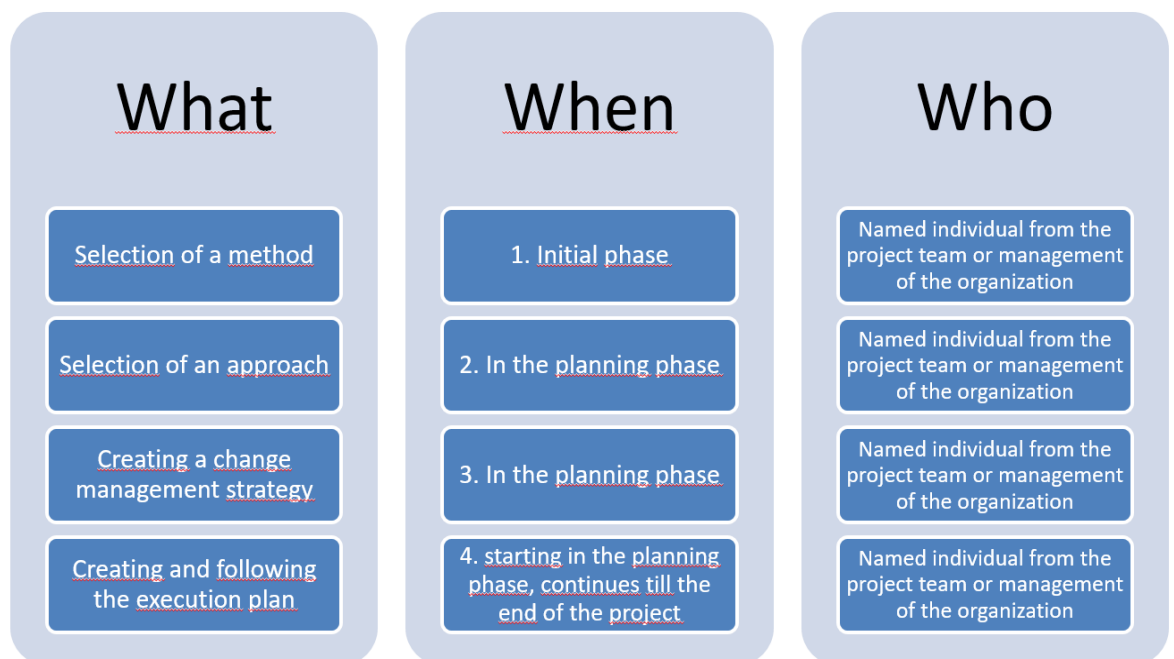


Figure 14. Final proposal.

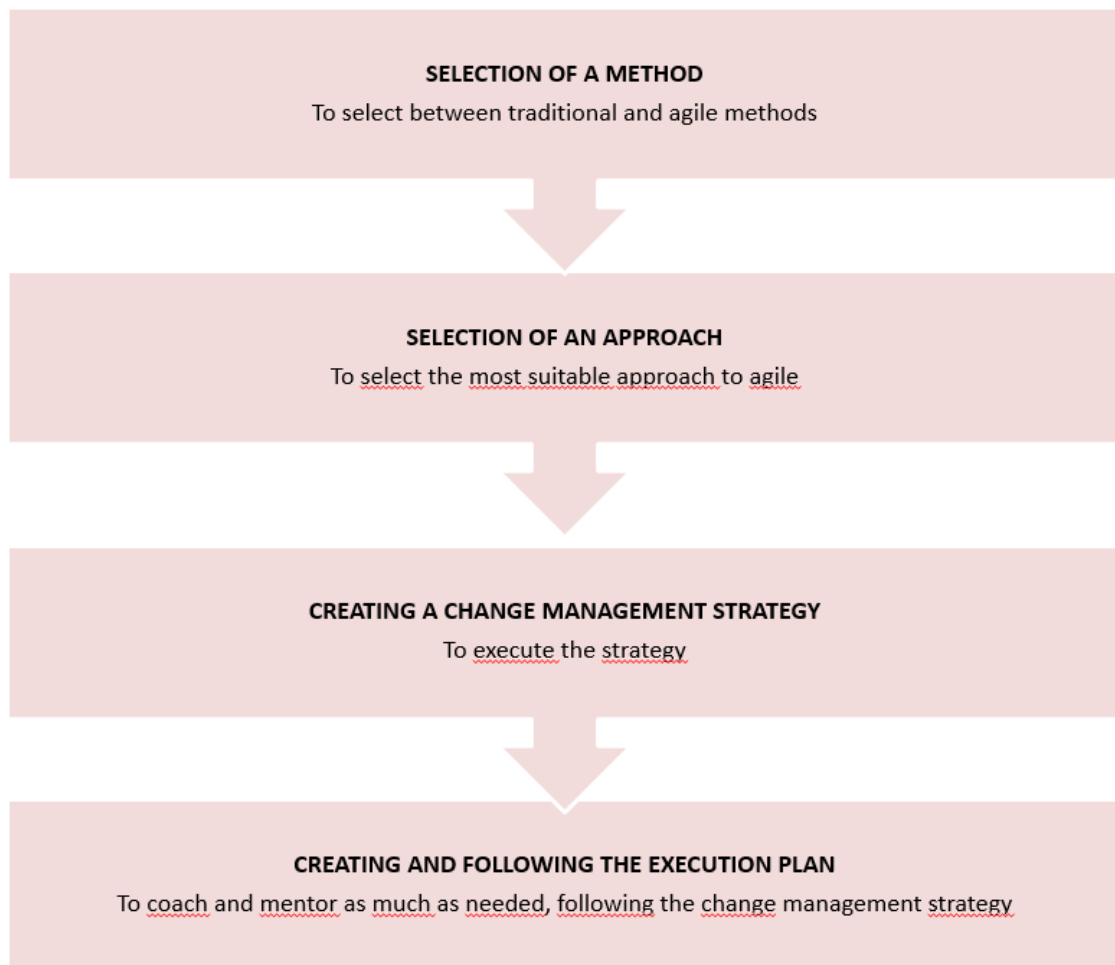


Figure 15. Final proposal.

	What		Who	When	Why
Action	Sub-actions	Preferred method to use	Responsibilities	Timing	
Selection of a method: to select between traditional and agile methods			Responsible person is named individual from the project team or from the management of the organization. To succeed, the person responsible requires sufficient knowledge of the organization.	Initial phase	To find out whether the pre-conditions of agile are met
	The organization is more people-centric than process-centric	Agile			
	The organization is more process-centric than people-centric	Traditional			
	The management style is more collaboration-oriented and responsive than control-oriented	Agile			
	The management style is more control-oriented than collaboration-oriented and responsive	Traditional			
	The knowledge management of the organization is more tacit than explicit	Agile			
	The knowledge management of the organization is more explicit than tacit	Traditional			
	The teams are self-organizing	Agile			
	The teams are not self-organizing	Traditional			
	The communication in the organization is informal and continuous	Agile			

	The communication in the organization is formal and rare	Traditional			
	The customer will likely be involved and actively participating	Agile			
	The customer will unlikely be involved and participating	Traditional			
	The project cycles will be guided by features	Agile			
	The project cycles will be guided by tasks and activities	Traditional			
	The evolutionary-delivery model will be used	Agile			
	The life cycle development model will be used	Traditional			
	The team members will be in same location	Agile			
	The team members will be in different locations	Traditional			
	The teams are encouraged to continuous learning	Agile			
	The teams are not really encouraged to continuous learning	Traditional			
	The project planning will be continuous	Agile			
	The project planning will be upfront	Traditional			
	The required documentation will be minimal	Agile			
	The required documentation will be substantial	Traditional			

Table 6. Checklist – selection of a method.

	What		Who	When	Why
Action	Sub-actions	Preferred agile approach to use	Responsibilities	Timing	
Selection of an approach			Responsible person is named individual from the project team or from the management of the organization. To succeed, the person responsible requires sufficient knowledge of the organization.	Planning phase	To select the most suitable approach to agile
	The organization does not have any experience on agile	Localization, (tailoring)			
	The organization is experienced on agile	Adoption, (tailoring)			
	The organization is willing to accept essential changes	Localization, adoption			
	The organization is not willing to accept essential changes	Tailoring			
	The organization will use agile and traditional methods side by side	Localization, tailoring			
	The organization will use only agile methods	Adoption			

Table 7. Checklist – selection of an approach.

	What		Who	When	Why
Action	Sub-actions	Preferred change management strategy to use	Responsibilities	Timing	
Creating a change management strategy			Responsible person is named individual from the project team or from the management of the organization. To succeed, the person responsible requires sufficient knowledge of the organization.	Planning phase	To consider what kind of change management strategy would be the most suitable
	The organization is willing to execute changes by the experts only	Expert			
	The organization is willing to execute changes by the management only	Directive			
	The organization is willing to let the management and people affected to negotiate together	Negotiating			
	The organization is willing to let the people affected to participate	Participative			
	The organization is willing to do compromises regarding how and what is to be done	Negotiating			
	The project is more technical	Expert			
	The organization is willing to accept a slower tempo	Negotiating, educative, participative			

	The organization is willing to execute changes fast	Directive, expert			
	The organization is preferring people supporting the change	Educative, negotiating, participative			
	The organization is ready to face major change resistance	Directive, expert			
	The organization is willing to learn and grow up, individuals including	Participative, educative			
	The organization is willing to invest resources and accept higher costs	Educative, participative			

Table 8. Checklist – creating a change management strategy.

	What		Who	When	Why
Action	Sub-actions		Responsibilities	Timing	
Creating and following the execution plan	Sub-actions		Named individual from the project team or management of the organization	The execution plan is created in the planning phase and the follow-up continues till the end of the project	To ensure smooth and controlled progress when transforming to agile, to coach and mentor as much as needed by following the change management strategy
	Plan resources	Name the driver-team and responsible person			
	Identify goals	Define the goals in detail			
	Identify risks	Identify the possible risks in detail			

	Create the execution process with sub-tasks	Identify needed training			
	Create a communication plan	Plan how/when/who will communicate and to what audience			
	Set the control points	Agree the scope, schedule, costs			

Table 9. Checklist – creating and following the execution plan.

The final proposal is trying to take all the previously introduced aspects into account to offer a useful checklist. The final proposal is telling who, what, when and why certain actions needs to be done.

8.1 Selection of a method

There are several things that organizations and individuals should be taken into account when planning to go agile. At first, a careful consideration which one, traditional or agile method would be preferable, should be done. Comparison between these two different methods should always be done case by case and understand the unique features in every project. There are cases where agile is not suitable at all despite of all the benefits it is offering. When doing the comparison, also the characteristics of the organization are crucial; some organizations are more traditional and rigid, having a lot of bureaucracy. It can be extremely challenging or even impossible to bring agility to organizations like this.

There must be a named individual responsible for the selection of a method; responsibility on selecting a method cannot be shared. Naturally, it is essential that responsible person is co-operating with other stakeholders and if needed, also consults subject matter experts, but he or she is responsible that the decision will be done appropriately and on time. Without a responsible individual who is having sufficient pre-conditions, there is an increased risk that this step will be done carelessly or ignored totally.

Also support from the management is needed; the way the support is needed is depending on the situation, but a minimum requirement is principled support. Sometimes also financial support may be required. Selection of a method is a big decision that should

not be done without a support from the management. Despite of a good plan, the first mistake is already done if responsible person with management support is not pointed out.

8.2 Selection of an approach

After careful consideration and selection of the method, desired approach should be defined. As introduced in earlier in the literature review, there are roughly three alternatives to select from; tailoring, localization and adoption. When selecting the approach, all aspects must be considered realistically, from the project and organizational point of view. One major thing impacting to the selection of the approach is the former experience on agile or the lack of it.

As in the first step of the proposal, selection of a method, also selection of approach requires an individual responsible with managerial support.

8.3 Creating a change management strategy

A change management strategy should be created by considering all known and common challenges, meaning management-, organizational-, people-, process and technology related aspects should be considered. The creation of a change management strategy must be done in the planning phase, after the method to follow and the approach has been chosen, before the actual project starts. As explained in the literature review, first the most suitable change management strategy approach to achieve the desired change needs to be defined. When defining the strategy, all aspects of the change must be taken into consideration; the organizational culture, the scale of the change, expected change resistance, schedule, budget and risks of the change.

The successful creation of a change management strategy requires also a named person who is in charge. Especially in this stage, the management support is crucial due to fact that changes may touch all aspects of the organization and have a significant impact on its customers as well.

8.4 Creating and following the execution plan

An execution plan is needed, together with the active follow-up. It is crucial to plan in detail how the actions will be executed; the plan itself is not enough but it needs to be followed-up as well. There must also be resources enough to execute the planned actions.

As with previous step, deep and sustainable support from the management is important. The management is also needed to provide sufficient resources and finance to secure the implementation of the execution plan.

9 Discussion and conclusions

9.1 The credibility of the study

The thesis is not built around a case company but done from a common point of view. Though the amount of people interviewed is not much, it was obvious that the answers and opinions were starting to be repetitive, hence there was not more interviewees involved. When considering the results of this study, it needs to keep in mind the pre-conditions, such as geographically location; since this study was done in a small country as Finland, it is obvious that the sizes of the projects are minor meaning that the use of agile is different than globally. In addition, the way agile methodology is used, is also depending on the organization. Some organizations are more agile-oriented than others and therefore better aware of the possible pitfalls. Out of the five interviewees, three of them were working as consultants at the time of the interview discussion; this is also a fact worth to notice since consultants may have different kind of possibilities to impact their customers' way of work and especially the way they are adopting agile and doing all the pre-work.

During the proposal validation, the developer commented that the outcome of this thesis is probably serving best Finnish people due to fact that the current state analysis was done based on interview discussions with Finnish people and the assumption that the use of agile methodologies is not yet very advanced in Finland. This is a useful view when considering the credibility of the thesis.

When considering the facts mentioned above, it can be said that the study is credible enough but the pre-conditions needs to be kept in mind. If a similar study would have been done in another location or in a selected case company, the results may have been a bit different. However, the issues identified in the current state analysis are matching to the preliminary literature in a high-level.

9.2 Conclusions

It was really educating to draw-up a study like this; the topic is near to my heart and I have been really interested on agile methodology and luckily have had the opportunity to use that in practise. I had originally a totally another topic, suggested by my employer of that time. I found this original topic to be too wide and it was difficult to seize that, hence I decided to do my thesis without a case company and select a topic that really fascinates me most. That was at the same time a really good decision but it also felt difficult to do the thesis without a case company supporting in a background, knowing there is nobody particularly ordering a study like this. Still I think the outcome of the thesis – a proposal how to overcome agile issues, in a form of a checklist, is valuable and useful for the companies planning or going agile.

References

Agilemanifesto.org, (2001). Manifesto for agile software development. [online] Available at: <http://www.agilemanifesto.org/iso/en/manifesto.html> [Accessed 16 Sep. 2017].

Agile alliance (2016). 12 Principles behind the agile manifesto. [online] Available at: <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/> [Accessed 16 Sep. 2017].

Balaji, S. & Murugaiyan, S. (2012). Waterfall vs V-Model vs agile: a comparative study on SDLC. International journal of information technology and business management, [online] Volume 2(1), pp. 26-29. Available at: <http://jitbm.com/Volume2No1/waterfall.pdf> [Accessed 9 Sep. 2016].

Bhuvaneswari, T., S Prabakaran, S. (2013). A Survey on Software Development Life-cycle Models. International journal of computer science and mobile computing, [online] Volume 2(5), pp. 263. Available at <http://www.ijcsmc.com/docs/papers/May2013/V2I5201384.pdf> [Accessed 9 Sep. 2016].

BBC Bitesize (2017). Software development cycle. [online] Available at: <http://www.bbc.co.uk/education/guides/z8n3d2p/revision> [Accessed 24 Mar. 2017].

Conboy, K., Coyle, S., Pikkarainen, M. and Wang, X. People over process: Key people challenges in agile development, pp. 2-3. [online] Available at <https://ulir.ul.ie/bitstream/handle/10344/639/2010-Conboy-People.pdf?sequence=2> [Accessed 9 Sep. 2016].

Denning, S. (2015). Agile: The world's most popular innovation engine. Forbes, [online] Available at: <http://www.forbes.com/sites/stevedenning/2015/07/23/the-worlds-most-popular-innovation-engine/#4920e2202d4c>

Gandomani, T., Ghani, A., Ziaei, M. and Zulzalil, H. (2013). Obstacles in moving to agile software development methods; At a Glance. Journal of computer science, [online],

p. 620-623. Available at: https://www.researchgate.net/publication/237077450_Obstacles_in_moving_to_agile_software_development_methods_At_a_Glance [Accessed 2 Sep. 2016].

Gandomani, T., Ghani, A., Sultan, A. and Zulzalil, H. (2012). Towards comprehensive and disciplined change management strategy in agile transformation process, pp. 2346-2349. Researchgate, [online], Available at: https://www.researchgate.net/profile/Taghi_Javdani_Gandomani/publication/262976245_Towards_Comprehensive_and_disciplined_Change_Management_Strategy_in_Agile_Transformation_Process/links/0a85e53980a84558a6000000.pdf [Accessed 2 Sep. 2016].

Ghilic-Micu, B., Mircea, M. and Stoica, M. (2013). Software development: Agile vs. traditional. *Informatica economica*, [online] Volume 17(4), pp.70. Available at: <http://www.revistaie.ase.ro/content/68/06%20-%20Stoica,%20Mircea,%20Ghilic.pdf> [Accessed 9 Sep. 2016].

Gothelf, J. (2014). Bring agile to the whole organization. *Harvard business review*, [online]. Available at: <https://hbr.org/2014/11/bring-agile-to-the-whole-organization> [Accessed 9 Sep. 2016].

livari, J. & livari, N. (2010). The relationship between organizational culture and the deployment of agile methods. *Information and software technology*, [online], p. 511. Available at: http://robertfeldt.net/courses/agile/livari_2011_ist.pdf [Accessed 2 Sep. 2016].

Kotter, J. (2011). Change management vs. change leadership – what’s the difference? *Forbes*, [online] Available at: <http://www.forbes.com/sites/johnkotter/2011/07/12/change-management-vs-change-leadership-whats-the-difference/#4a040b318ec8> [Accessed 9 Sep. 2016].

Kotter international. 8-Steps process for leading change. [online] Available at: <https://www.kotterinternational.com/8-steps-process-for-leading-change/> [Accessed 2 Sep. 2016].

Lichtenberger, A. (2014). What IT service management can learn from agile manifesto (and vice versa). [online] Available at: <http://blog.itil.org/2014/08/allgemein/what-it-service-management-can-learn-from-the-agile-manifesto-and-vice-versa/> [Accessed 16 Sep. 2017].

Lockitt, B. (2014). Change management. [online] Available at: <https://web.archive.org/web/20061010032354/http://nt6139.vs.netbenefit.co.uk/pdf/CHANGEMANAGEMENT3t.pdf> [Accessed 2 Sep. 2016].

Moczar, L. (2013). Why agile isn't working: Bringing common sense to agile principles. CIO, [online]. Available at: <http://www.cio.com/article/2385322/agile-development/why-agile-isn-t-working--bringing-common-sense-to-agile-principles.html> [Accessed 9 Sep. 2016].

Rohweder, T. (2016). Strategy, management and leadership.

Schwaber, K. & Sutherland, J. (2013). The scrum guide – the definitive guide to scrum, pp. 3-7. [online] Available at: <http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf> [Accessed 9 Sep. 2016].

Scrum.org (2016). What is scrum [online] Available at: <https://www.scrum.org/resources/what-is-scrum> [Accessed 9 Sep. 2016].