Universiteit Leiden

ICT in Business

Software Development Methods in Large Regulated Environments.

Name: Coen Vermeij

Student-no: s1351370

1st supervisor: Dr. Christoph Stettina

2nd supervisor: Prof. Dr. Aske Plaat

Master Thesis

Leiden Institute of Advanced Computer Science
(LIACS) Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

ABSTRACT

Organizations have difficulties to find the right configuration for agile in large organizations. Current models do not help because (1) old models are too bureaucratic, and (2) there is an agile sweet spot in which agile can give maximum benefit to the organization. However, The sweet spot of IT projects is so specific that it is hard to maintain when the organization does more than just software development.

This thesis deals with the balance between the organizational environment and the agile context. In order to achieve this, we are doing research to find the balance, between on one hand, the adaptation of the organizational environment and on the other the level of adopting and adapting the context of the agile method. Thus we raise the following question: *"How to achieve the right balance of adapting the organizational environment on one hand and adapting the software development method on the other, in order to create the biggest benefit for the organization?"*

To answer this question we did an multiple in-depth case study where we looked into four large regulated organizations. Based on observations during multiple rituals in context and semi-structured interviews, spread over multiple organizational levels, we collected qualitative data which supported our findings on the research matter.

From this research we can conclude due to the organizational environment and the need for adaptation of the software development method, that the balance for each organization is different. The factors table as is defined in this thesis can help organizations to better understand the balance of organizational environment and agile in context.

ACKNOWLEDGMENTS

First of all I would like to take this opportunity to thank all those involved in supporting this thesis research project. All four of organizations, which could see the benefits of this study and the support they gave me during this research.

Secondly we would like to acknowledge all the individuals who took the time to sit with us for our data gathering. We also want to acknowledge all the individuals who were kind enough to show us around the organizations and let us sit in during rituals.

Furthermore I would like to thank the supervisors from the university for their guidance and support during the research thesis.

TABLE OF CONTENT

1. INTRODUCTION

This chapter will give the reader an introduction of this research thesis. In this chapter the reader will find a short summary on the motivation of this research. Furthermore this chapter describes the research objective, the associated research questions and scope and will end with the relevance of this research.

1.1 Motivation

In today's market, organizations are heavily investing in agile software development methods for their company. Small organization seem to have it easier implementing and using agile software development methods than larger organizations. Based on internet research and papers published about the subject, this seems to be correct. But why is this the case? Why do larger organizations struggle more to implement agile software development methods? There are success stories on larger organizations successfully implementing agile software development methods. What sets these organizations apart from the other ones?

When looking at organizational structures from Mintzberg (2006), we see that there are different organizational structures, where some could potentially support agile methods easier than others. Is this the case? Is it so, that based on the structure of the company, agile software development methods would be easier to adopt within an organization?

Many organization today look like Machine Bureaucracies, as defined by Mintzberg (2006), however, we live in a knowledge society these days. Which means that we have specialist and professionals working in our organizations. Can a Machine Bureaucracy support these workers?

If we take the table of Mintzberg (2006) on the structures on organizations , we see that organizations and organizational structures resembling the professional bureaucracy deals with specialists. A lot of organizations, especially larger organizations, struggle to adopt or adapt agile development methods in their organization due to the structure of the organizations. the Professional Bureaucracy dealt with specialists, but in a single individual and not in teams. The knowledge in an professional bureaucracy is not management among the specialists. (Mintzberg, Structures In Fives Designing Effective Organizations, 2006). Would a professional bureaucracy more fitting to adopt agile than other structures?

The Professional Bureaucracy from Mintzberg (2006), can be characterized by the following characteristics, as described and cited by Mintzberg;

By standardizing coordinating mechanisms, the organization allows for decentralization. The organization has highly trained specialist in its operational core. They are the working force of the organization. These specialist are given a considerable amount of autonomy, to do their work as they please. This is because the specialist are the ones with the knowledge regarding their work and they know how to best manage that work. The amount of autonomy in the operational core of the organization, is why this operational core is so large.

There is little use for middle management, if they are no subject specialist themselves. In order to maintain a small piece of the middle management, there are some managers with subject knowledge to support decision-making within the organization.

The professional bureaucracy functions best in an environment which is both complex and stable. As complexity demands a skilled specialist with subject knowledge and skills which is learned in extensive training programs. Due to the stable environment, these knowledge and skills can become the operational standard of the organization.

There are characteristics which should not interfere with this organizational structure. These factors are regulations and automation. Any of these characteristics could destroy the individual operator. Autonomy would be replaced by administrative or peer group influence. This will result in a shift of organizational structure to a different configuration.

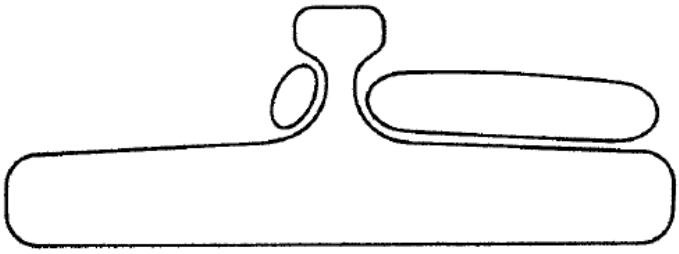


Figure 1 - Professional bureaucratie structure

Mintzberg's organizational structures, Professional Bureaucracy and Machine Bureaucracy, matched against each other to see the differences between the two structures (Mintzberg, Structures In Fives Designing Effective Organizations, 2006).

	Professional Bureaucracy	Machine Bureaucracy
Premiere coordination mechanism	Standardizing of Skills	Standardization of Work and processes
Central part of the organization	Operational Core	Techno structure
Design Parameters		
Specialization	Broad Specialty	Broad and deep specialty
Training	Lots of Training	Almost no training
Formulation and Behavior	Few formalization	Lots of formalization
Grouping of units	Functional and customer oriented	Functional
Size of units	Large at base (operational core) limited elsewhere	Large at base (operational core) limited elsewhere
Planning and Control	Low planning and few control	Action planning
Connection	Straight connections to management	Low connection possibilities
Decentralization	Broad decentralization	Limited broad decentralization
Functioning		
Strategic Top	External contacts, problem-solving	Regulating, coordination on functions, problem solving
Operational Core	Qualified, standardized work with individual autonomy	Routine formalized work with few input
Middle Management	Controlled by professionals, lots of specialized adjustments	Extensive and diverse, problem solving, supporting for broad structures.
Techno structure	Few	Extensive for formalization of work
Supporting Staff	Extensive support by professionals	Extensive to improve workplace safety
Management structure	Insignificant, except in the operational core	Significant through entire organization
Influence Regulated Systems	Insignificant, except in the operational core	Significant through entire organization
Informal communication	Significant in management	Discouraged
Work units	Only in management	Insignificant especially in operations
Decision Model	Bottom-Up	Top-Down
Situation Factors		
Age and size	Variant	Old and large
Technical Systems	Not regulated or advanced	Regulated but not automated, not advanced
Environment	Complex and Stable	Simple and stable
Power	Power is with the professionals	Technocratic and with influence from outside

Figure 2 - (Mintzberg, Structures In Fives Designing Effective Organizations, 2006)

1.2 Research objective

This research is inspired by companies which are struggling with implementing or adopting an agile development method in their organization. What these companies have in common is that they are regulated in some way in their development process (Cawley, Wang, & Richardson, 2010). In this context, regulated means that the desired outcome, is delayed due to some factor in which the organization is restricted. These restrictions often depends on the organizational structure (Mintzberg, Structures In Fives Designing Effective Organizations, 2006) and the context in which the regulations are placed, in both the organizational environment and the context of the software development method.

These regulations can be voluntarily places by the organization to control the flow of the process, but these regulations can also be involuntary (Martens, 2014). Meaning that the regulations are placed by the organization or a controlling entity on to a department which is struggling with the regulation.

In some organizations, software projects require the approval of certain people or roles in the organization, in order to begin or continue the development process. This is because there are regulations in the organization set by the business, to which IT and development needs to obey. These regulations will differ in every organization, depending on the vision and roadmap of an organization. These restrictions has as a result, that the development process of software is restricted in their freedom. Freedom in the development process is what agile development promotes (Hajjdiab & Taleb, 2011). This is a shift from traditional development to agile development.

Despite what most people think, the development process of a software project is not just the actual coding of a program. The entire process of working out an idea, analyzing possibilities and identifying features is just as much part of the development process (Vo, 2007). Most development processes start with the gathering of requirements from users and the business (CMS, 2008).

For an organization to make the most use of a development method such as agile, the organization needs to know in what context the development method is created, what the regulated constraints are and how this effects their development process. For instance, if an organization enforces a restriction stating that a new software project needs to be pre-defined by business cases and scenario planning's, this has impact on the actual development of the project. By limiting the freedom of the developers, the team can work less agile, since this is a part of the agile method (Beck, et al., 2001).

1.3 Research questions

This research seeks balance between regulations in business and the freedom of development tools such as agile development methods. For this research the following research question has been identified:

- *Research Question: How to achieve the right balance of adapting the organizational environment on one hand and adapting the software development method on the other, in order to create the biggest benefit for the organization?*

The reason why we define software development methods in the question, is because we want to research, if, in some cases, it would be better to drop agile methods for a more traditional waterfall approach. We want to research if regulations in the business restricts software development in some manor and what the consequences would be.

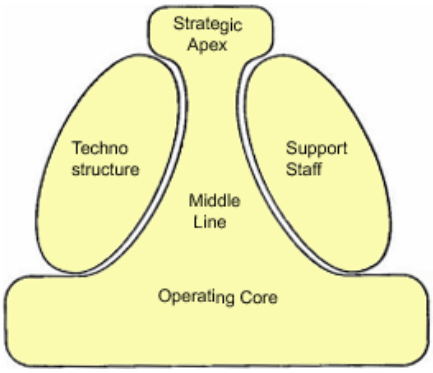
To gain the desired answer for the main question of this thesis, we have identified the following sub-questions which would provide the researcher with the desired data. The sub-questions are:

- Sub-question 1: *What organizational factors, in large software developing organizations, are different when looking at agile and traditional perceptions?*
- Sub-question 2: *What are the organizational factors, within large regulated organizations, which are perceived to stand out?*

Software development methods are used to deliver benefits to the organization. The most popular software development method currently used, are agile methods. These Agile methods states continuous delivery and early return on investment among other potential benefits. Within an regulated environment, it appears be more difficult to ensure that the method can deliver said benefits. Is this the case? What would these factors be?

1.4 Research scope

Henry Mintzberg has written a model in which he divided an organization in different sections. In this model, the IT department is usually a supporting role in an organization, which puts it in the Support Staff section (Mintzberg, Structures In Fives Designing Effective Organizations, 2006). Because IT is becoming more important, organizations which rely heavily on IT products, have organized their IT department in the Operating Core of the company. These organizations supply IT services. By doing so, the organization commits itself to the IT department and supply them with sufficient resources, since it's their core business (Mintzberg, Structures In Fives Designing Effective Organizations, 2006).



Because IT only fulfills a supporting role in the organization, they are more restricted, then when the IT department would operate in the Operating Core of the organization, because IT is not seen as their core business (Fitzgerald, Stol, O'Sullivan, & O'Brien, 2013).

Figure 3: Organizational visualization

In this research, we will only look at the process of new software development. From this research, we will exclude software projects such as changes and bug fixes for reasons. The first reason is because these software projects tend to be smaller. Smaller software projects are less vulnerable for regulations, than new software development. Also bugs can hamper the performance of the organization, this can be a reason to green light a bug fix even though it may be expansive. In this case it is a necessity to execute a bug fix, this is why we exclude bugs from this research.

Even though the development method of agile seems to be the most popular method these days, it is not the main research objective of this research. This research will identify what regulations typically arise in organizations and will look at how these regulations influence the implementation or use of a development method. This means that this research is not limited to promote agile in regulated organizations.

1.5 Motivation of relevance / importance

Agile is a popular development method (Larman & Basili, 2003). Due to its success, a lot of companies that are developing software, are attempting to adopt and adapt agile in their organization. In their perspective, agile offers freedom and flexibility, customer satisfaction and more control of their projects. However, agile teams have implications with the structure, process and culture of an organization (Stettina & Hörz, 2015). There are companies, where the adoption of agile takes a long time. In some companies, it is nearly impossible to adopt an agile method due to the amount of regulations that the organization needs to apply. This means that the difference between the software development context and the organizational environment is to large. This explains why for some organizations, it is easier to adopt agile, while for some organizations, it can become a struggle. Some organizations are even trying to define a hybrid form between agile and a waterfall method to get a fit between the software development method context and the organizational environment.

This study is relevant because is adds knowledge to the scientific literature. During our findings we were unable to find literature which relates to the issue of finding the right method with the certain organization. The existing literature describes what your organizational environment should be like when adopting a software development method. What the literature does not describe, is how you get your organization to fit with the context of the agile methods and what you should do with the challenges and obstacles the organization needs to overcome to get there. Though this research cannot describe the implementation of these methods. There is just too much difference between organizations. What this research can do is the identification of regulating factors in organization, and describe how to deal with these regulations which are likely to influence with organizational environment in relation to the software development context.

1.6 STRUCTURE OF THE THESIS

This research thesis consist of 6 chapters. The first chapter is used for the introduction and placing the research in context.

In chapter 2, we will discuss the theoretical framework. The framework consists of a literature study on the topic matter and will give insight in the current software development methods and the context of the these methods.

Chapter 3 is describing our research methods and design. We will discuss why we used a multiple in-depth case study as our research design and how we selected our case organizations.

Chapter 4 describes the results we find during our research. we will discuss what types of data we collected and how we collected the data.

Chapter 5 is the discussion chapter. In this chapter we will discuss our findings and further elaborate on our findings. This chapter consist of 6 sub chapters each with their own theme of data which we found and will be discussing.

Chapter 6 will describe the conclusions that we concluded during our research, based on the results and discussion chapters.

2 THEORETICAL FRAMEWORK

In this chapter the researcher describes what is already known about the research material.

2.1 Software Project Management in context

2.1.1 Adoption

The adoption of development methods differ due to the fact that each development methods is different. A traditional development methods will pose different challenges than the agile development method. It also matters if the organization has done agile before and is only adopting a new agile method, or the organization has always done software development with a traditional software development method and are adopting an agile method now (Nerur, Mahapatra, & Mangalaraj, Challenges of migrating to agile methodologies, 2005).

Each of these adoption scenarios will pose challenges. Organizations are often unaware of the amount of change that is required from the organization (Laanti, Salo, & Abrahamsson, 2011). Adoption of a software development methods is highly dependable on the context of the method and the environment in which the software development method will be adopted. When adopting an agile software development method, not just the environment of the organization needs to be taken into account. Both the development team have to adopt the software development method practices and the organization (Hoda, Kruchten, & Noble, 2010). Even customers should adapt to this method, because they have a representative participating in the software development teams.

Adoption of software development methods depends on organizational characteristics (West & Grant, 2010). These characteristics come from organization but also from the environment that the organization is in. These characteristics and environment also determine in what degree the organization needs to adapt the software development method.

2.1.2 Adaptation

Adaptation of software development methods is a necessity, however to some degree. There is no one-size-fits-all software development method, so tailoring a method and adapting it to suit the organization and its environment is a must. However taking in account that these methods are created in a certain way that they deliver the highest benefits. This is the method in context. The context is the environment in which the method delivers the highest benefits. Altering and adapting the method would derive benefits (Hoda, Kruchten, & Noble, 2010).

2.1.3 Contextualization

Software development methods are without doubt successful within their own contexts. The context can be seen as the environment in which the software development method is used. For each software development method there is a certain context in which the efficiency for the methods is at its most. This context differs for each method and it can be hard to identify if the environment of your organization resembles such a context.

Agile development methods are among the most used development methods at this moment (West & Grant, 2010). While success stories of agile adaption and execution are known, rumors of organizations which struggle to adopt a working agile methods also exist. This is due to the fact that the organization is not aware that there is a specific context in which agile operates best. The designer for development methods had a certain context in mind for their method, this is also known as the “sweet spot” (Kruchten, 2013). It is the context in which agile is most suited and will work most efficient. Projects which are executed within the sweet spot benefits most from the advantages of the methods (Hoda, Kruchten, & Noble, 2010). Using agile development methods outside of this sweet spot doesn’t mean that agile will not work. However this does mean that agile “out of the box” will not deliver the expected results. This means that adaptation of the agile development method is requires to operate at a desired level.

Kruchten (2013) discuss that there are two levels of context; Organizational level and Project level. The organizational level factors relate to the organization's vision on software development, while the project level is more focused on project management.

These contexts should be seen as the environment is which the development method should operate. Therefor the context should match the existing environment in which the organization, adopting the development method, operates. Meaning the existing environment should support the context in which the development method is placed. If there is no fit between the context and the environment the development method would encounter difficulties.

The sweet spot of agile is according to Kruchten (2013) the following:

Figure 4: Kruchten (2013) - the Agile sweet spot (in blue)

2.1.4 Regulated factors

There are regulated factors within these certain environments, which make the adoption and adaptation of agile in an organization a lot more difficult. This would mean that the environment or organizations differs much from the context of software development methods, resulting in the loss of software development method benefits.

McLeod and MacDonell (2011) have identified four main factors that influence software development methods practice outcomes;

1. Project content (Product / Project)
2. Development processes (Process)
3. People and Actions (People, Roles and Responsibilities)
4. Institutional context (Organization and culture)

We created a table to identify the factors per category mentioned.

Organization	**Process**	**Product/Project**	**People**
Governance	Flexibility	Method adaptation	Project teams
• Team autonomy • Lack of commitment by the business • Organizational prerequisites and conditions • Organizational maturity • Business Model	• Organization flexibility • Predefined requirements	• Stable architecture • Stable environment	• Roles and responsibilities • Formation and autonomy • Involvement of stakeholders • Dedicated teams • Team distribution
Culture	Communication	Resources	
• Adaption of/by employees • Cultural habits • Specialist Culture	• Communication between development (IT) and business	• Resources and financial allocation	

Figure 5: Table identifying factors per category

Organizational factors

Governance

According to Moe et all (2009), organizational control is a hurdle which could indicate that software development team do not get adequate *team autonomy*.

This organizational control can be related to *lack of commitment by the business*. Lack of sufficient *team autonomy* can result in an situation where the software development team members do not see the point of committing to team goals, due to the fact that they are the specialist and the business do not respect their judgment and decisions.

Kettunen & Maarit (2008) agree that *lack of commitment by the business* happens in large scale organizations. However, Kettunen & Maarit (2008) also mention that, due to *organizational prerequisites and conditions*, the organization has reasons not giving the development teams the desired level of autonomy.

Culture

Nerur et al (2005) argues that culture is a hard thing to change within an organization and that culture exerts considerable influence on the business. This relates to the amount of organizational compliance the business can impose on the *adaption of employees*. Where project managers previously enjoyed authority, will they be able to relinquish this authority to support a new software development method? Letho & Rautiainen (2009) agree that culture is one of the most difficult things to change in an organization. A single change in an employees work process can cause resistant, making the *adaptation by employees* more difficult.

Process factors

Flexibility

In traditional organizations, processes and procedures are strongly in place. With the arrival of agile, organizational flexibility has become an issue. *Organizational flexibility* has become important for the implementation of agile software development. If the software development method is based on agile methods, but the organization is not based on agility, this business will not keep up with the rapid changes in development. According to Hoda, Kruchten & Noble (2010) the need for this flexibility is acknowledged but often not addressed due to *predefined requirements* from the organization.

Communication

In a traditional organization*, communication between business and IT* was based on progress reports, which development send to keep the stakeholders informed about progress (Hardy, 2006). Organizations are still stuck in traditional habits and processes, where the management is not in touch with IT development. The result of this is lack in communication between management, but also between the business and IT (Letho & Rautiainen).

Product factors

Resources

Funding IT projects, according to Thomas & Baker 2008, is done the same way, as we would do with a traditional method like waterfall. It is part of the annual budget meeting. This means that the budget for IT is set prior to the start of the upcoming fiscal year. To identify the IT budget, the IT agenda for the entire year should already be known or is already set. The business expects that the entire list of preset projects, according to business cases, will be delivered within the allocated time and budget (Thomas & Baker, 2008).

People factors

Project teams

Depending on the selected software development method, project teams should have different *roles and level of responsibility*, same goes for the *formation and autonomy* of project teams (Meso & Radhika, 2006). Modern software development project teams are easier to govern than traditional project teams. This is due to the greater flexibility, transparency and the involvement of stakeholders (Ambler, Scaling agile software development through lean governance, 2009).

Making the teams with a fixed setup, their relationship will strengthen, capabilities and knowledge will grow, resulting in more accurate planning (Cheng, Jansen, & Remmers, 2009). According to Moe, Dingsøyr and Dybå (2009) shared resources (*non-dedicated teams*) is a difficult way of working, due to the fact that developers can be called away to work on other software. This can even be software which was delivered at an earlier stage. According to Sutherland et al. (2007) distributed team, always have disadvantages in compared to collocated teams and will never preform on the same level, yet organizations maintain over shore development teams.

2.1.5 Regulated Environment

For this research, we need to explain what is meant with regulated organizations and regulated environments, since it is not a term with a fixed definition which is used in literature. “Regulated organizations” or “Regulated Environments” are terms which are used in different ways and in a different context in the literature. Most of the times “Regulated Organizations” is related to organizations in the financial, FDA, Healthcare or industrial sector (Martens, 2014) (Ambler, The Agile Scaling Model (ASM): Adapting Agile Methods for Complex Environments, 2009) (Serguei , 2004). Some papers go as far as adding nuclear and automotive organizations to the term “Regulated Environment” (Fitzgerald, Stol, O'Sullivan, & O'Brien, 2013). Regulated environments are therefore strongly associated with safety and security. We see regulated organizations or regulated environments as an organization, which are regulated by entities that the organization itself has no control over. This can relate to an organization, but also to departments within organizations. In organizations where IT is seen as a supporting role, IT is usually heavily regulated by the business. For the purpose of this research, we will define this department also as organizations as they usually operate next to the business instead of with the business.

Within a regulated environment, there is a responsibility from the IT department to the business, to supply proof of compliance. This compliance consist of financial compliance, organizational compliance and architectural compliance. Meaning, is the IT department in its right to spend the money, is it in the organizations best interest and does it fit within the current infrastructure of the company. Even though infrastructure is part of IT responsibility, to regulate the freedom of expansion, the business want the regulated environment to comply with business rules (Cawley, Wang, & Richardson, 2010).

2.2 Software Project Management and Development methods

In this research, we will only consider the more common and most used development methods. Taking into account that there are other development methods, outdated methods are considered irrelevant in this study, because they will not be adopted by modern companies in today's business market. This means that in this research we will use the following development methods; Waterfall development, Rapid Application Development and Agile Development. Even though the Waterfall development method is discussed to be outdated by some, a lot of companies still use this method. The reason why we discuss these methods is to determine what development methods exist and how they are adapted by the market for more efficiently. We want to know how these development methods relate to organizational environments.

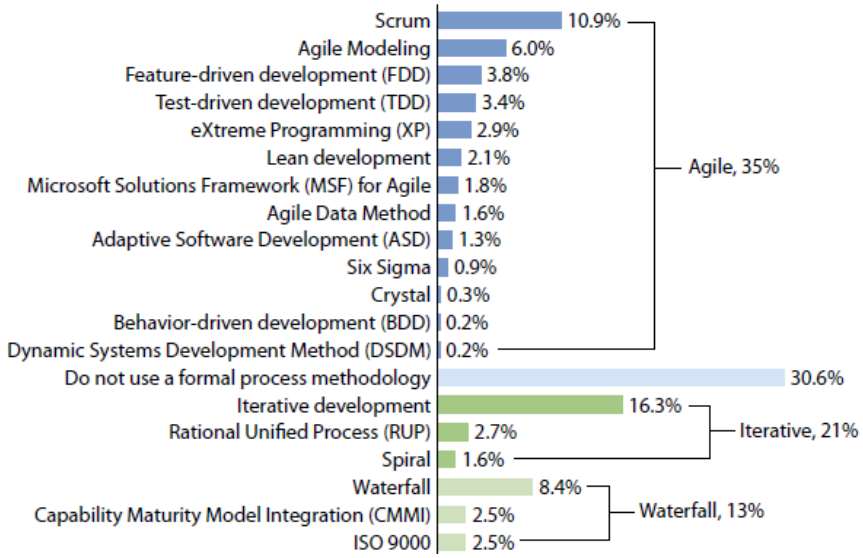


Figure 6: Forrester/Dr. Dobb's Global Developer Technographics® Survey, Q3 2009

With a shift in software development methods, a shift in the reasoning and understanding how and why projects work, has also happened. In traditional software development methods, the requirements where fixed, meaning that a software development project would have fixed requirements but resources and time were just estimations. This could result in an expensive project which lasted a long time and perhaps with outdated requirements.

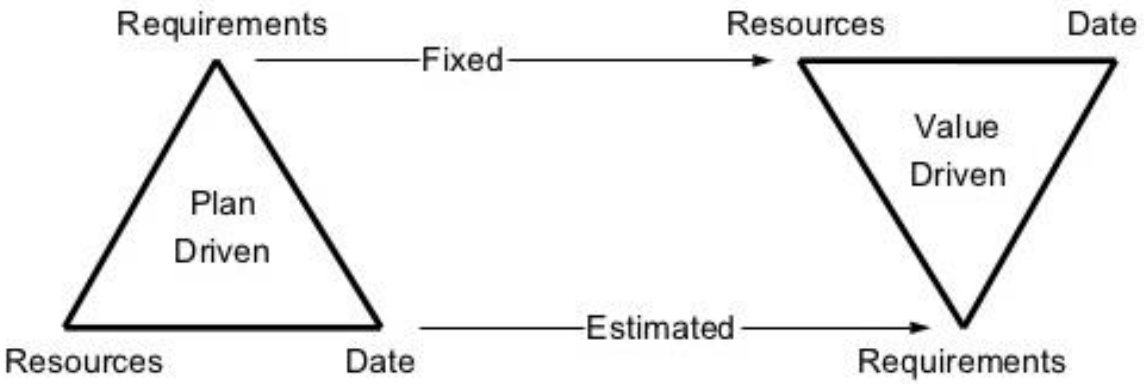


Figure 7: Leffingwell 2011

With agile software development methods, the resources and time are fixed, making the requirements dynamic. This results in a project where the developers can discuss how many resources and time they need, to deliver parts of the project. The traditional methods and the more recent software development methods will be discussed so that the reader can understand the shift in reasoning.

2.2.1 Waterfall development

The waterfall model is based on a sequential design process. This model is used for software development and is known for its linear development. The first appearance of the concept of a defined Waterfall model in the year 1970 (Royce, 1970) (Petersen, Wohlin, & Baca, 2009). Even though the model seems outdated, it does not appear to be forgotten any time soon (Boehm B. , 1988). A lot of organizations still use the waterfall method for development, even though researchers are focusing on the importance of agile (Petersen, Wohlin, & Baca, 2009) (Larman & Basili, 2003).

The waterfall method in most cases consists of 5 phases; Requirements, Design, Implementation, Verification, Maintenance. In the original version of Royce however, the phases; Construction and installation were among these phases as well, making it a 7 phase model. Varied modified waterfall models are built to address some of the criticism on the pure waterfall model.

The waterfall model states that the next phase can only be started when the previous phase is reviewed and verified. The result of these verification can implicate that the project need to return one or some phases when down the stream complications occur (Royce, 1970).

The philosophy of waterfall is that making sure requirements and design are right early on in the development process, saves money and time later in the process. This means that a lot of attention is spent on the first phases of the development process. Each phase needs to be as complete as possible before going to the next phase. (Verma, Bansal, & Pandey, 2014) States that the waterfall method can best be used if the requirements are clear and fixed, the technologists stable and when the technology used, is understandable and static.

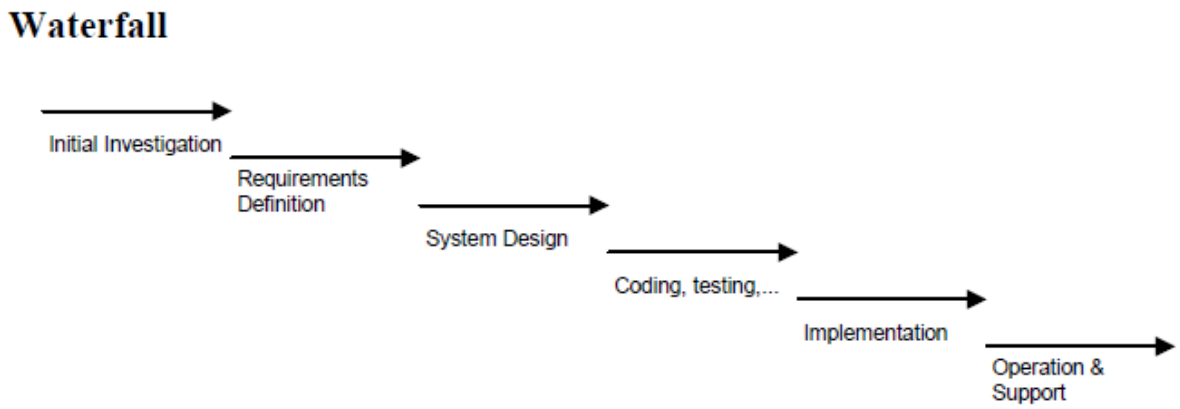


Figure 8: (CMS, 2008)

2.2.2 Rapid Application Development

Rapid Application Development (RAD) is an umbrella term for development models which are built on the principle of rapid development and prototyping. RAD approaches software development differently than the traditional waterfall method. RAD focusses more on the development and spend less time on planning and startup documentation such as predicting requirements. A RAD development method is prototyping, which is a way to gain faster working software.

In prototyping, the developer builds prototypes and discus the prototype with the user and customer. When the user and customer gives their approval, the developer builds further upon the prototype to create the next set of features. Prototyping result in risk reduction, because required features are built in early and problems are identified at an early stage (CMS, 2008). Also requirements can easily be adjusted during development due to the availability of a prototype. Prototyping does not work in every environment. When the software is critical for the core process of an organization, RAD would not be the best solution (ProjectManagement, sd). Programs in a regulated environment like health care, where people's lives depend on the working of software, prototyping is not a solution. During prototyping, the users are taking the prototypes into production environments.

Although Prototyping is a strong development method in the right situations, it also has some disadvantage when used in a less suitable environment (Sommerville & Kurkovsky).

Because there is very little documentation, managing the project can be hard, due to the lack of pre-defined goals and milestones. Due to the lack of documentation, requirements and features are not documented. This can cause contractual problems, because no project end is specified. The software is as good, as the developer who is building it. This makes RAD a large commitment from users and developers.

Organizations in a dynamic, complex or changing environment, have a hard time defining a consistent set of system requirements. This means that a waterfall model would not be suited for these sort of organizations. RAD is suited for these kinds of environments because of user and customer involvement (Sommerville & Kurkovsky).

Prototyping

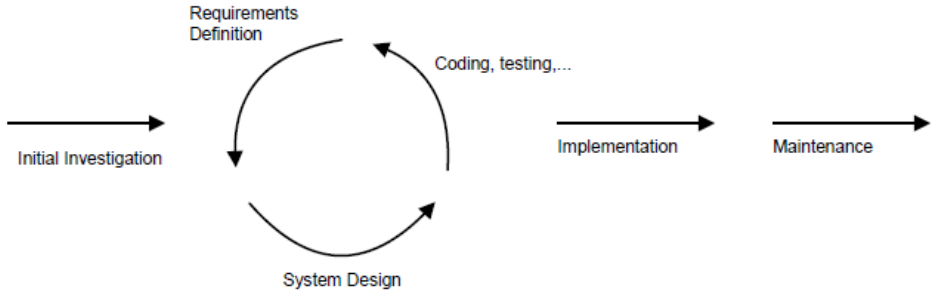


Figure 9: (CMS, 2008)

Spiral Development is another method based on RAD and is a combination of Linear and Iterative development. Spiral development focusses on risk identification and tries to minimize risk by segmenting a large project into manageable chunks. Each of these chunks, go through the same 4 stages of the spiral model. These stages are: Determining objectives, Identify and resolve risks, Development and Test and Planning of the next iteration (Boehm B. , 1988). Each cycle begin with identifying all stakeholders involved in this chunk and define what they requirements are. Each cycle will result in acceptance from these stakeholders, confirming that the chunk delivered what was agreed on (CMS, 2008) (Boehm B. , 1988) (Boehm & Hansen, 2000).

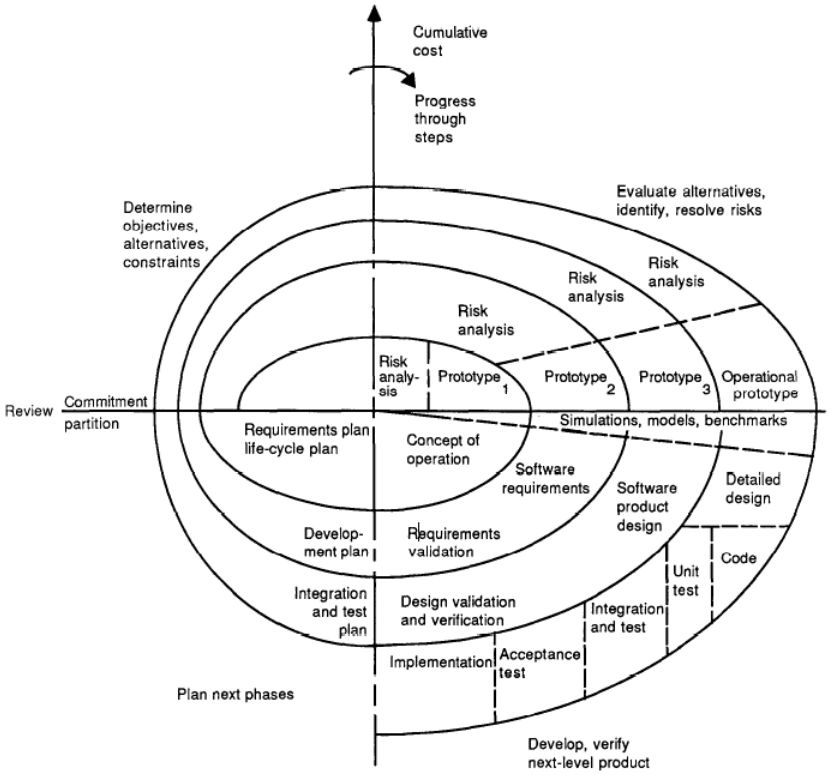


Figure 10: (Boehm B. , 1988)

2.2.3 Agile Software Development

Agile Software Development is an umbrella term for software development methods which are based on iterative and incremental development. Agile development is done with collaborative self-organizing teams. These teams operate in small numbers and regularly deliver small pieces of software during a fixed amount of time. These pieces combined makes the software project. There multiple agile methods in existence, but not every method is appropriate in every environment (Abrahamsson, Warsta, Siponen, & Ronkainen, 2003).

Agile is based on four values and twelve principles (Beck, et al., 2001). The values;

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

The twelve principles can be found in appendix 1.

In agile software development, the project is divided in smaller projects which can fit in short cycles, which agile defines as sprints. In each of these sprints a certain functionality is produced. All these sprints combined create the final product. Agile seems to be successful in both small and large organizations (Ambler, The Agile Scaling Model (ASM): Adapting Agile Methods for Complex Environments, 2009). What these organizations have in common is that they have tailored an agile method, to fit their own unique environment (Ambler, The Agile Scaling Model (ASM): Adapting Agile Methods for Complex Environments, 2009). Not all organizations are successful in adopting an agile method in their organization. Not all organizations are able to adopt an agile method in the desired way, that it can deliver the desired benefits.

According to Meso & Radhika, there are seven best practices in agile methods (Meso & Radhika, 2006).

1. Frequent releases and continuous integration
2. Need for frequent feedback
3. Proactive handling of changes to the project requirements
4. Loosely controlled development environment
5. Planning kept to a minimum
6. Enhancing continuous learning and continuous improvement
7. Emphasis on working software product

For agile development methods, implementation will be different in every organization attempting to adopt an agile method. This is due to the fact that every organization, environment and context (depending on the method) is different (Nerur, Mahapatra, & Mangalaraj, Challenges of migrating to agile methodologies, 2005).

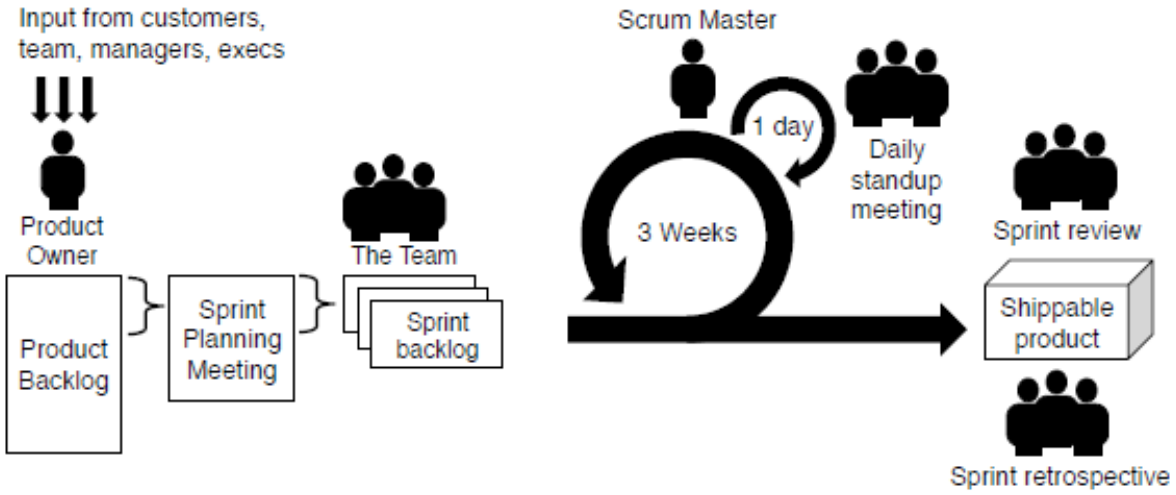


Figure 11: (Fitzgerald, Stol, O'Sullivan, & O'Brien, 2013)

2.3 Agile Governance and Organizational Models

Because agile is one of the most popular development method at this point, we will discuss the known frameworks and practices, to discuss how these practices interact with regulated and complex environments.

The Scaled Agile Framework is a publicly available framework for applying lean and agile practices in larger organizations.

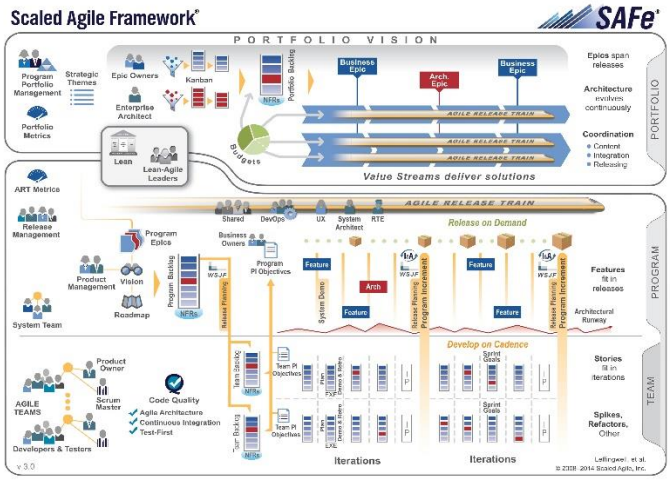


Figure 12- Scaled Agile Framework - Appendix 2

The Scaled Agile Framework, SAFe for short, describes in their framework, that it has found a balance between the agile sweet spot and the organizational environment. According to SAFe, their framework is a preset framework, which needs to be adjusted to fit within the environment of the organization which tries to implement agile, using this framework.

For SAFe to work, all the entities, represented in the model, need to fulfill a role within this environment. If one of these entities is not committed to their role or simply not available in the organization, the SAFe framework, just like other frameworks, would not work. This framework shows a good overview of an organization and the interrelated connections that all the roles in the organization has. Not all practices can work within all organizations, this is why SAFe mentions that the organization should only adopt and adapt practices which can work within their organization (Leffingwell, Scaled Agile Framework, 2014).

3 RESEARCH METHOD AND DESIGN

For this research project, we will start with a literature review of the topic material. Literature review will be done on scientific papers published. To find these papers, the internet search function of Google Scholar will be used, together with the University library catalogue. This literature review will give the researcher the theoretical foundation on which this research thesis is created, which is called the theoretical framework.

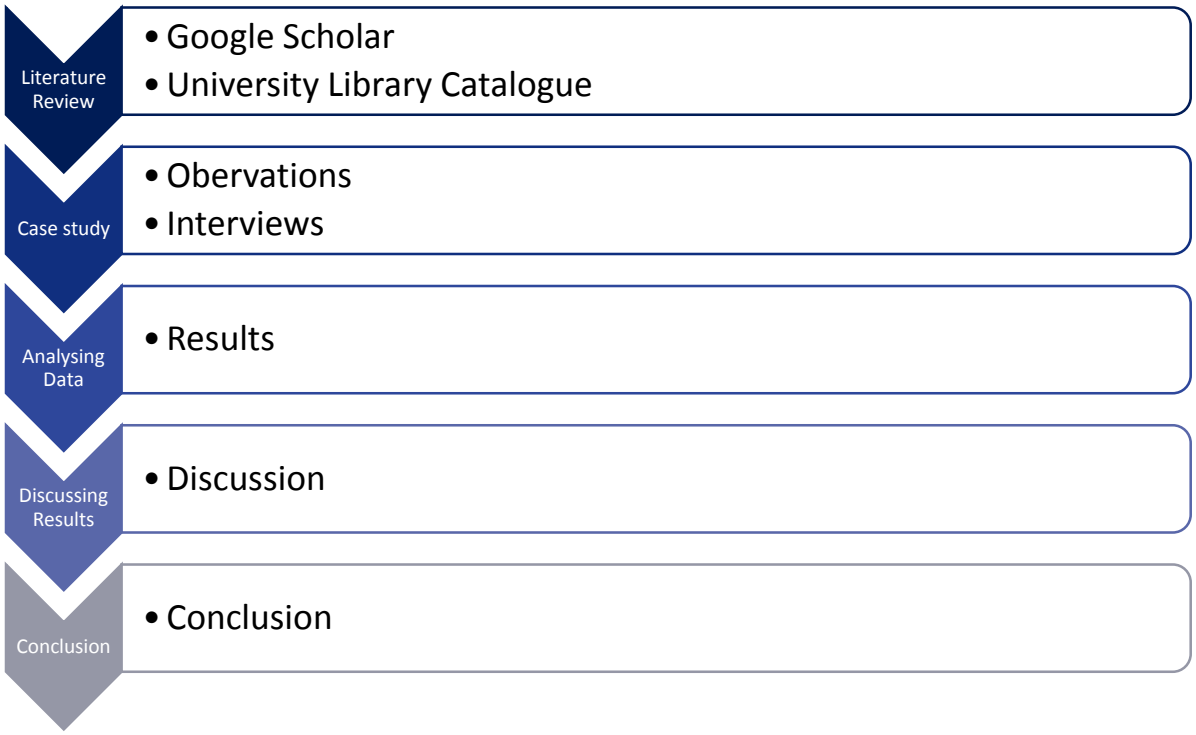


Figure 13: Research design

3.1 Literature review

For the literature review we have searched for papers, articles and books which could give us more information, to answer our research questions. The literature we have searched for must meet some requirements. We want relevant papers for this research, one of our requirements was that the research papers regarding different subject must meet a certain age or publish date. Papers regarding agile adoption must be newer than the year 2000 to deem the paper relevant. Papers regarding agile development methods should also meet this requirement, while papers regarding waterfall and rapid application development methods don't have to meet this requirement. These methods are older and we used earlier by the business. However, even though these methods are older, they are still for our research, because we want to know if they could be more suited to work in a regulated environment.

The sources from where we gather our theoretical knowledge must also be validated. Because we use the internet to search for the literature, the sources must be validated. Articles places on regular websites are usually not validated and are in most cases written from the perspective and emotion of the writer and does not contain verified facts. This is why we only use literature materials from verified sources.

To gain the literature required for our study, we used keywords to search. The keywords which we used are: Regulated environments, Agile, Adaptation, Software development methods, adoption, and contextualization.

We combined these keywords to gain results such as: "agile in regulated environments". The make sure that we were able to find the most relevant papers, we used synonyms for our keywords to broaden the search. E.g. complex and dynamic environments instead of regular.

3.2 Multiple In-depth case study

A multiple case study was chosen for this research, based on Robert K. Yin's "Case Study Research". The reason why we chose a multiple in-depth case study is, because we want to see how different organizations, which are working in different environments, react to agile transitions. These organizations have to work in the same organizational context, meaning that they are highly regulated.

The case studies should supply the researcher with data which we use to offset our findings from the literature review. To get the desires data from the organizations, semi-structured Interviews and direct observations are used.

The interviews supply input from multiple roles, spread through different levels of the organizational. For each organization 4 – 5 persons are selected for an interview. The goals is to interview at least one person per one of the four levels in the organization:

- Directors/ higher management
- Agile coaches
- Business
- Operational staff

The interview guide is attached in appendix 3. Everyone who had been interviewed got the same interview guide as a guideline during the interviews. However, if conceived as more beneficial, the researcher can deviate from the interview guide if the interviewee feels that he can supply more information that way.

Next to the interview guide, each interviewee is asked to fill in an table, which is conceived from out literature study. In this table we identified agile and traditional factors, divided over the four categories:

- Structure
- Development process
- Project content
- Culture

This table helps us to identify what the perceptions of all the interviewees are, in relation to agile and traditional factors. Because this table is the same for everyone taking the interview and the results are always a binary choose, the results are easily measured. This lets us identify if there are any potential deviations are. The table is supported to be filled in twice. Once for the interviewees to show their current perceptions on these factors and once to show their desired perception of these factors.

The reason why we ask for the two different perceptions is to identify how far the organizations are to perceived in their agile transition. The interviewees can mention that some factors need to be placed more in the agile context or even the other way, the factor need to be more traditional.

We will be using the results of this table to visualize the perceptions of the different roles in the organizations. Due to anonymity, the results will always be displayed as the average of the organizations. With the exception of the interviewed agile coaches. The agile coaches play a large part in the agile transition of these companies and their perception of the agile state in the organization is interesting to offset against the rest of the organization.

3.3 Case Selection Strategy

As mentioned, besides the literature review, multiple case studies have been done for this research. Four organizations which are positioned in an regulated environment, have been selected for this study. These organizations are active in the following sectors:

- Commercial organization
- Banking organization
- Service organization (working with governmental organizations)
- Governmental organization

We have selected these organizations because all of these organizations are of considerable size and are currently in an agile transition. Within each of these organizations, agile coaches are or were present to guide the transition.

The selected organizations allowed us, during the time we spend on site, to do observations, which are conducted at different agile rituals. The goal of these observations are to observe how the organization is managing agile within the organization. Several rituals were observed using the direct observation method, where the researcher was present, but not participating with the rituals. These rituals are observed to identify if the organization is using started rituals or if the organization has self-created rituals.

4 RESULTS

In this chapter we will identify the results we encountered from our research. We will first mention what kind of data we collected and how we got this data. After this we will summarize the organization which participated in our studies, after which we will talk about how the data related to the organizations.

4.1 Data

The data which we collected consists of 17 interviews, spread over 4 organizations (5-4-4-4). The time spent per interview was 1 hour on average. During the interview we asked the interviewees to fill in our factors table. Pending on the level of understanding of the agile method, the time spend on filling out the factors table was between the 10 to 20 minutes. The factors table generated discussions on the perceptions of agile against traditional factors. This data is added to the interview data.

The factors table is used to gain insight in the perceptions of the interviewees in relation the current transition toward agile methods. The factors, used in the table are the following;

Agile Factors	Traditional Factors
Dedicated teams	People are part of multiple teams and projects
More cross functional teams	Specialist line organization
More flexible governing rules	Management dictates rules
Decentralized decision-making	Centralized control
Informal communication	Formal communication
Shared workspace	Separated workspace
Co-located developers	Developers located anywhere
Small team size	Large team size
Extended support staff (also knowledge exchange)	Basic support staff (administration, HR, finance, safety)
Focus on updating systems	Focus on maintenances (legacy systems)
Iterative process	Linear process
Value driven - Adaptive approach	Plan-driven - Predictable approach
Incremental delivery (continuous)	All in once delivery
Direct Value	Future Value
Test every Iteration	Test at project completion
Interactive input from customer - High client involvement	Predefined user requirements - Low client involvement
Low refactoring cost	High refactoring cost
Allow change in scope	Deliver Fixed Scope
Teams are integral part of estimation	Management provides estimates
Continuous improvement each iteration cycle	Lessons learned improvement after project
Learning through coaching, workshops, games	Formal education only
Small project size	Large project size
Low amount of documentation	High amount of documentation
Customization and in-house development	Off-the shelf software
Leadership and collaboration	Command and control
Collaborative environment	Competitive environment
Management commitment and collaboration	Management as controller, not involved in content
Management is encouraging improvements	Management is dictating ways of working
Transparency of goals, resources and progress	Unclear goals, resources and progress
Strategy and vision is embraced by the entire organization	Strategy and vision is something for top-managers
Decisions are made based on metrics	Decisions are based on opinions
Boards and comities act as teams	Boards and comities are political playgrounds

Figure 14: Factors table

The third data source comes from the observations which were conducted at the organization. We were present in the organizations to observe certain agile rituals. During these rituals we focused on the execution of the rituals and the behavior of the people. At least two agile rituals were observed per organization. a total of 16 observations were done, spread (unevenly) over the 4 organizations.

4.1.1 Interviews

The data we gathered during our Interviews consist of answers to the questions of our interview. Each participant got the same interview questions which makes the answers possible to compare. This data supplies the research with their experience and perspective of their organization in relation to the agile transition.

From the total of 17 interviews conducted, 4 of the interviews were at organization A, B and C. Organization D supplied us with 5 interviewees. Within each organization, we conducted this interview with different roles in the organization. We always divided the interviews over 4 organizational levels;

- Director/management
- Business
- Agile coach
- Operational staff

Each interview was recorded to be transcribed at a later stage. Each recorded interview was processed by the researcher and the answers of the interviewees were transcribed in the interview guide. On average the transcribed interview guide is 4 pages. The interview contains general questions about the interviewee, 15 semi structured interview questions and a small sentiment analysis. The interview guide can be found in appendix 3.

4.1.2 Table data

During our interviews we used the factors table which resulted in the agile against traditional factors data. This data is structured as a binary value. This makes the data comparable and can be used to make graphs for visual support for our findings.

In the factors table the interviewee noted down two values. An X mark for the interviewees current perspective of the agile or traditional factors and an Y mark to indicate the desired perspective of the interviewee. The factors table is filled out by the same people who participated in the interviews.

4.1.3 Observations

The data we collected from our observations will be used to determine what kind of rituals and practices are used in the organizations. During the observations, the researcher is present and is writing down every event or action which is happening during a ritual. Each record made during the observations has a timestamp, rounded to the nearest minute. A total of 16 observations (1490 minutes of formal observations) were done, with 3 observations in both organization A and B. 8 observations were done in organization C and 2 observations were done at organization D. In order to be able to compare rituals, we wanted at least one ritual with we observed over all the organizations, to be the same. In all the organizations we attended a Spint Planning meeting, which is an Agile ritual. A large part of the results from an observations is culture. Culture also the aspect which is clearly noticeable during these observations. Since culture is also one of our measure categories in the interview table, we are able to compare the perceptions of the interviewees with this data.

4.2 The organizations

We selected 4 organizations, divided over multiple sectors. Below is an table with the summary of the context of the organizations.

Organization	A	B	C	D
Size/ employees	500+ internal 20000+ external	80.000+	350+	3.500+
Size of IT department/ teams	40+ with 2 agile teams	160 agile teams throughout the organization	10 agile teams in the organization	Significant IT organization with multiple agile teams
Industry/sector	Logistics and production	Banking	Governmental service supplier	Governmental
Start of transition	2 years	4,5 years	2 years	1 year
Initiation of transition	Bottom-up	Top-down	Bottom-up	Top-down
Method of transition	Started a new project in an agile approach and implemented agile throughout the IT department	Gradually implementing Agile in parts of the organization	Started a large new project with an agile approach	Implementing agile in smaller parts of the organization and not all in once
Culture	Directive and regulated Culture	Formal culture, command and control	Informal culture	Static and Traditional governance
View on IT	Directors see IT as Supporting, while IT sees that they are far more important for the organization	The organization can't exist without IT	Because they are an IT service provider, IT is their core business	IT support the processes and are positioned accordingly
Regulation	Highly, forced by management	Highly, forced by organization	Organization still traditional	IT infrastructure is highly regulated
Interview / Roles	Agile Coach / Scrum master Product owner Line-manager Developer	Agile Coach IT Director Team manager Line-manager	Agile Coach / Scrum master IT Architect Line-manager Developer	Agile Coach Scrum master Product Owner Line-manager Developer
Observations	Daily Standup Sprint Planning Retrospective	Daily Standup Sprint Planning Sprint Review	Daily standup Sprint Planning Sprint Review Retrospective Acceptance sprint meeting User acceptance test Stakeholder demo Refinement session	 Sprint Planning Scrum of Scrums

4.2.1 Case study organization A

Organization A is an commercial logistics and production company in the Netherlands. The organization has 500 employees working in the office and around 20.000 employees working outside the office. The information management department has chosen to implement agile software development methods for their in-house software development. the organization develops products which are used by both their own employees and their customers.

The agile transition to agile, in organization is based on a bottoms-up approach. The information management department has started the initiation and is currently in the transition of adopting agile in the organization. The information management department is seen as a supporting department in the organization.

4.2.2 Case study Organization B

Organization B is one of the leading Dutch banking organizations. Organization B has started 4,5 years ago to implement agile as a software development method in their organization. Because organization B is a large organizing (80.000+ employees), the agile implementation is executed in steps through the organization. The agile transition in this organization is still in progress. This means that the organizing is still learning how agile methods are best implemented in their organization.

Agile is implemented at different times in different parts of the organization. The management is committed to implement agile in the organization. This results in a top-down approach. IT is seen as a leading factor in this organizations.

4.2.3 Case study Organization C

Organization C is an service provider, working inside the Dutch government. In this organization, software is created for one of the Dutch ministries. Organization C is currently using agile software development methods for their software development. The organization currently has more than 350 employees.

Agile methods are only in use in the development departments of the organization. the IT department of this organization is seen as a leading part of the organization, due to the fact that they are an IT service provider.

4.2.4 Case study organization D

Organization D is an large Dutch governmental organization. This organization is trying to adopt agile practices for the time of a year now. Some external help is present in the organization. Because the organization of is significant size, the organization is adopting agile in just a part of the organization.

External help is guiding the transition together with some internal personal, together they are forming an agile task force. They are attempting to adopt agile, in an agile way in the organization. Starting with a smaller part of the organization and are slowly involving more and more people. The view on IT is this organization varies. IT is seen as an important part of the organization, however IT is also being restricted in some ways.

4.3 Organizations explained

In this we use graphs and charts as an illustrative way to support our results and findings. Below the graph or chart is the explanation of the visualization of the data. In this paragraph we identify the context of the organizations in a visual manner, supported with text for elaboration.

When we look at the results of the factors table, divided in categories, we can determine where the largest gap is, in the agile factors from current situation to desired situation. We want to know what, on average of the organizations, the largest gap is to be found in the overall culture.

Figure 15 visualizes the average of deviation in factors, all data of the organizations are added to a total and divided by the amount of questions per category to get and rational representation of the weight of the category. Taking the representation and dividing it by the total representation allows us to identify the relative percentage of the weight per category over all the organizations.

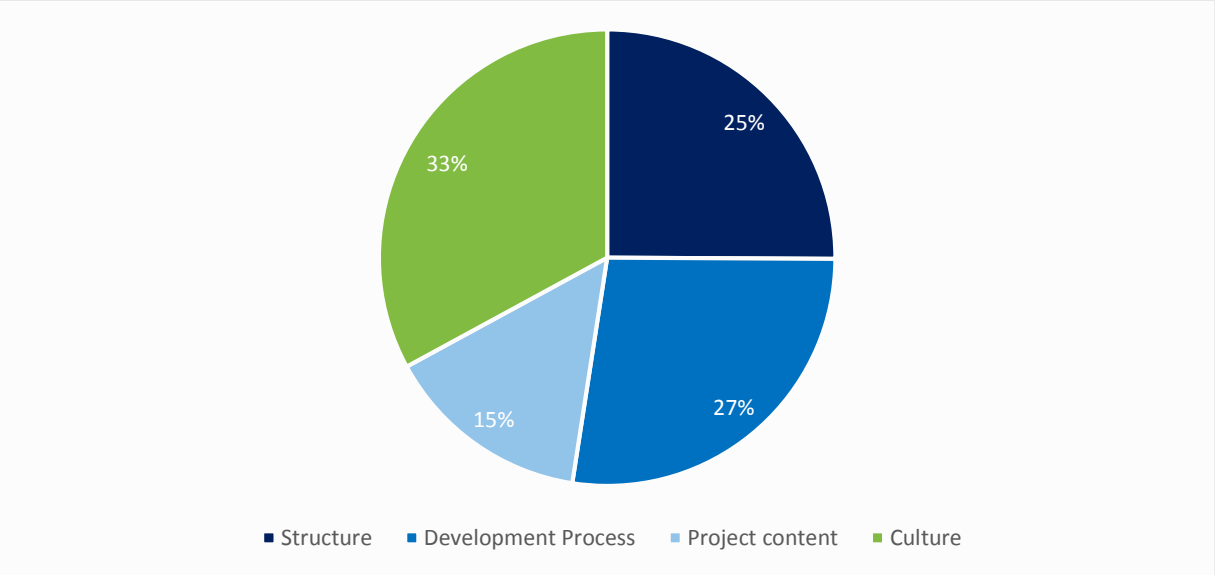


Figure 15: Average of deviation in factors

Organizations	A	B	C	D	Total	# questions	Representation	% out of 100
Structure	14	9	11	6	40	10	4,00	25,08
Development Process	8	10	16	14	48	11	4,36	27,36
Project content	2	1	2	2	7	3	2,33	14,63
Culture	18	6	11	7	42	8	5,25	32,92

Figure 16: Results from the factors table

Culture is where the most interviewees said that they want to move more towards agile factors. Culture is described in our questions as, how much regulations or freedom is there in the organization, how does management involve themselves and how is strategy and goals imbedded in het organization. Culture is also one of the most difficult aspects in an organization to change. However, not all organizations have equal need for more agile culture.

To determine the results in figure 17, we looked at the difference between the current agile factors and the desired agile factors. The difference between these factors is what we call the gap. The percentages displayed in the charts, represents the relative amount of increase in agile factors in the desired state.

Organizations	A	B	C	D		A	% per Cat of A	B	% per Cat of B	C	% per Cat of C	D	% per Cat of D
Structure	14	9	11	6	10	1,40	27,76	0,90	31,12	1,10	23,93	0,60	17,57
Development Process	8	10	16	14	11	0,73	14,42	0,91	31,43	1,45	31,65	1,27	37,28
Project content	2	1	2	2	3	0,67	13,22	0,33	11,52	0,67	14,50	0,67	19,53
Culture	18	6	11	7	8	2,25	44,61	0,75	25,93	1,38	29,92	0,88	25,63

Figure 17: Results from the factors table

4.3.1 organization A

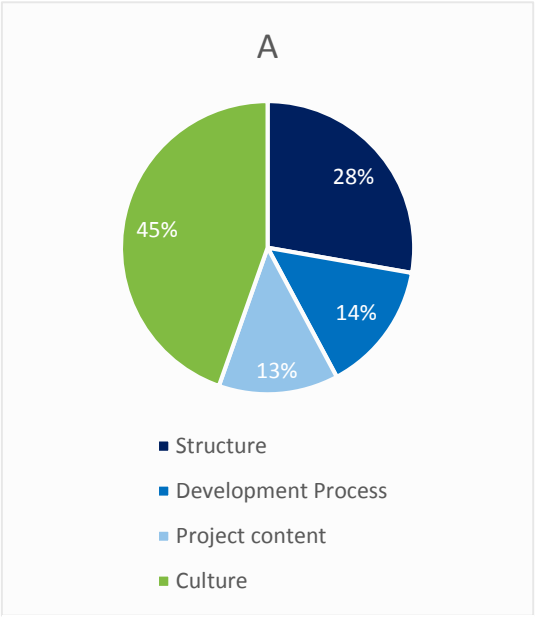


Figure 18: Gap organization A

In organization A we see that the desire for a more agile culture, is highly present. 45% of the total increase in desired agile factors were desired in the culture category. This would indicate that the culture within organization A is perceived to be more traditional, which would confirm out statement about the low agile culture in organization A. The second largest desire to improve, which stands out is the structure category. This would indicate that the organization is still trying to implement agile as a development method in the organizations. The structure category focus on the environment where the software development methods is operating in.

4.3.2 Organization B

When we look at organization B we see that the largest need for agility is located in the categories of Structure and Development Process. Based on this data and the observations in the organization, we conclude that the reason for this result is that the organization is busy with the transition in the organization. Just parts of this organization are adopting agile at this point. Which means that within the organization, agile coaches are still busy with supporting the employees making the transition to agile.

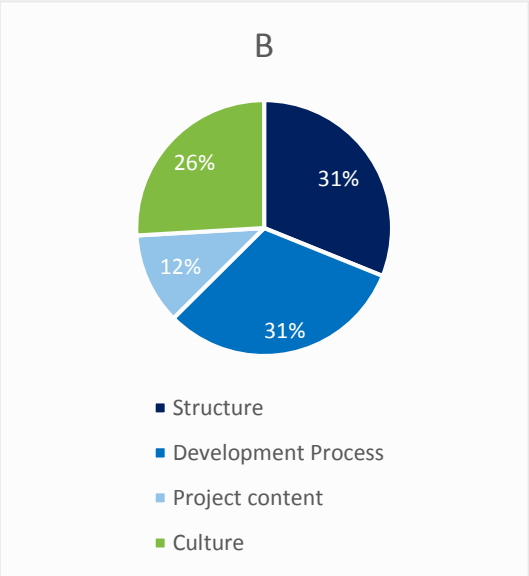


Figure 19: Gap organization B

4.3.3 Organization C

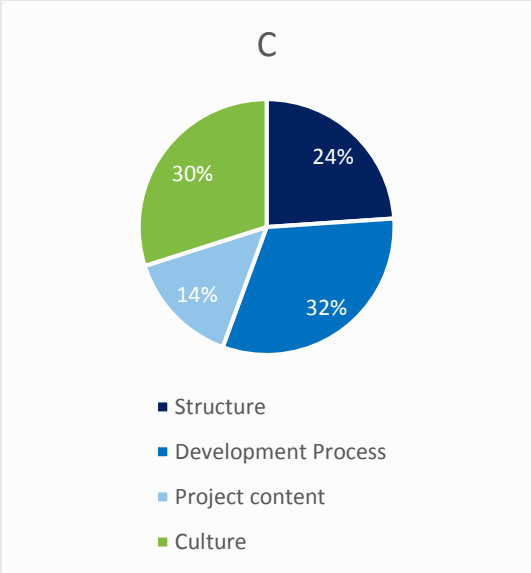


Figure 20: Gap organization C

Within organization C, we identified that both the Development process and the culture requires the highest amount of transition towards more agile.

This organization is the one where the difference between categories is the smallest. This means that the transition toward more agile work is more in balance in this organization that in the other organizations.

4.3.4 Organization D

Development Process describes the way that the organization uses the method, how they manage the method and who is participating. This category also stands out in organization D. When we look at culture, we see that this factor is representing 26% of the desired change. This makes this the factors which is most in balance.

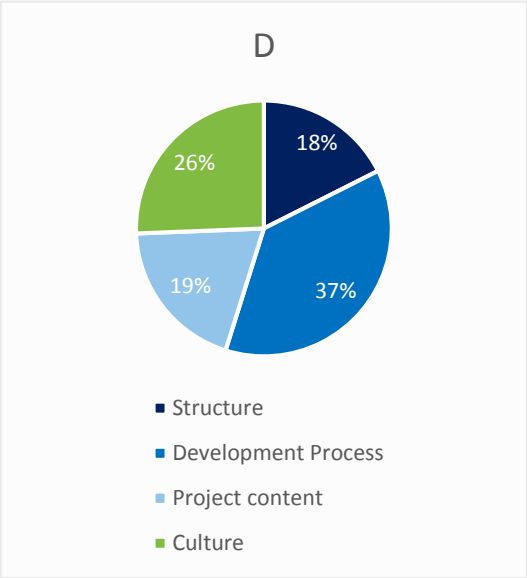


Figure 21: Gap organization D

4.4 Transition of agile methods

In this paragraph we describe the transition steps, which the organization used, to introduce agile in to their organizations.

4.4.1 Organization A

As suggested by their IT supplier, organization A started doing one project using an agile method. This project resulted in success. After this project, based on the results, agile became the new development method for the organization. This decision of implementing agile was located in the IT organization of the company, even though agile software development was introduced by the IT supplier. The organization adopted the agile method by:

- Getting the knowledge inside the organization by sending people to training.
- This knowledge was used to form agile teams and started executing the projects in an agile way.
- Agile coaches were appointed to guide the transition.

4.4.2 Organization B

Based on best practices from within other departments in the organization, a manager decided to look at agile methods. The Decision was made based on speed and quality of the development method, after which both IT and Business had been reorganized. The announcement was made that the organization would undergo an transition towards agile development methods starting on a certain date. In order to achieve this:

- New roles were introduced in the organization.
- Followed by Training sessions for the employees.
- Just before the transition date, agile coaches were assigned to guide the organization.

4.4.3 Organization C

Agile was chosen as the new software development method, based on the recommendation of a developer. Agile was used as trail for some smaller projects and was considered to work as an software development method. When a large project was announced with lots of uncertainties, agile was chosen as the method to use during this project.

- Communicated through the line management that agile would be the new method.
- Slowly learning by doing
- Agile knowledge was gathered by training and agile coaches were appointed.

4.4.4 Organization D

Organization D was under pressure from its environment to research the possibilities of agile software development. Because their product development realization is taking a long time, the desire for agile software development methods is increasing. The following transition steps were uses in this organization:

- Get the vision on agile equal among the management
- Create a change management method, based on maturity.
- Look for ambition in the organization. Try to identify people who are willing to be ambassadors for agile in the organization.
- Create an roadmap where the goals and time are specified.
- Start a small initiative within the organization and slowly expand this initiative to the rest of the organization.

5 DISCUSSION

5.1 Agile against Traditional factors

Within each organization, Agile factors are present and most of these organizations are using a pure Scrum method. From the Scrum method, they take almost all the rituals and execute these in their organization. we talked with different persons, with different roles, in different organizations. During our interviews we asked them what they perceive as agile factors in their organization. From this we can determine that all the organizations are currently in an transition from a traditional organization towards a more agile organization.

Figure 22 visualizes: (1) Perceived current state on a scale of agile vs traditional configuration. (2) Perceived desired state on a scale of agile vs traditional configuration. (3) Graph indicated desire to change within an organization.

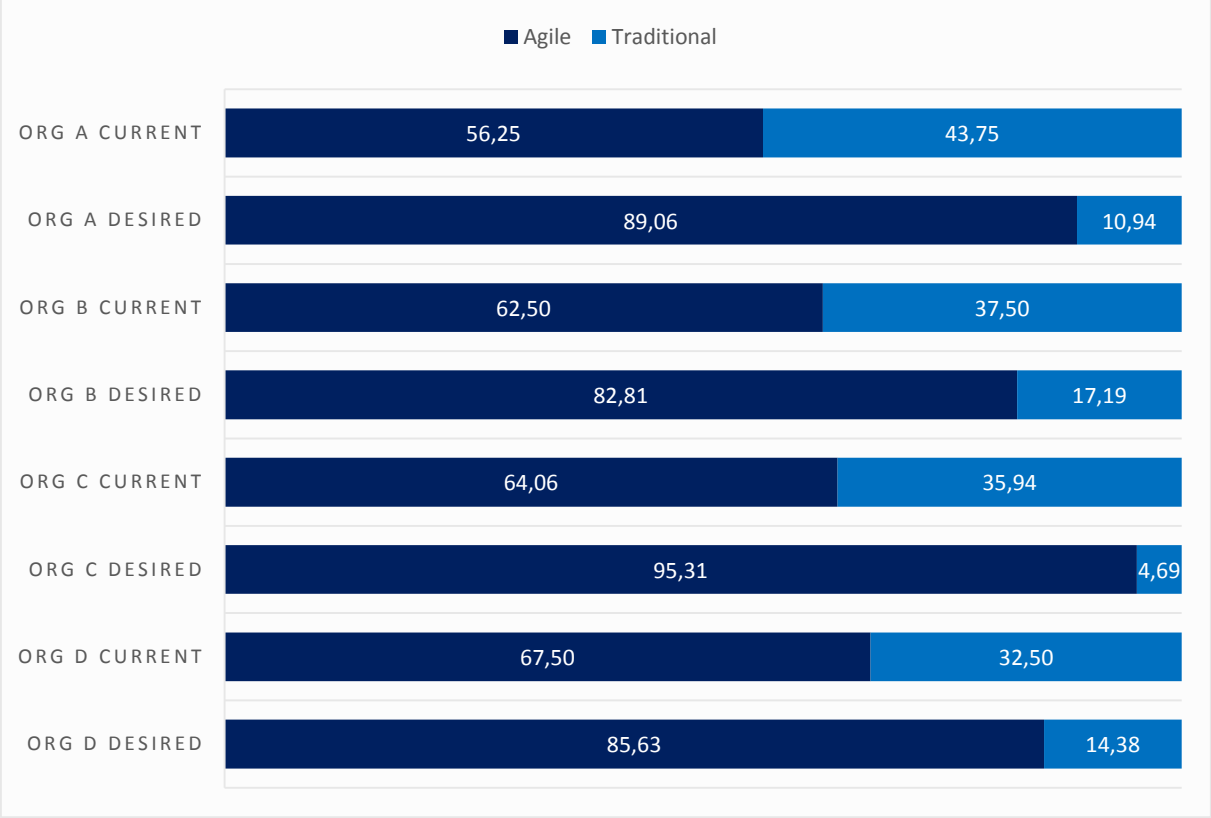


Figure 22: Perceived current and desired factors

Figure 23 visualizes the Gap between the current and desired outcome per organization. The gap is the difference between the current and the desired outcome according to the factors table data.

Gap Org A	Gap Org B	Gap Org C	Gap Org D
32,81	20,31	31,25	18,12

Figure 23: Gap between current and desired factors

In the graph above we see the current balance of the agile and traditional factors in the these organization. As the graph displays, organization A has the least agile factors present in their organization, based on perceptions of the interviewed employees. This means that there are still traditional factors present, which exist next to agile factors.

Organizations B and C have more perceivable agile factors in their organizations, where organization D has the most perceivable agile factors present. It is worth mentioning that Organization D has stated the agile transition just one year ago, which is the shortest time, compared to the other organizations.

During our interview, we also asked what the desire amount of agile factors was in relation to traditional factors. If we were to compare the desired results, we see that organization D is not the leading organization. These results show the desired situation in the organization, based on the perspective of the interviewed persons. Even though the results of all companies are different, none of the organizations desire a 100% full agile approach based on our identified factors.

Organization C, based on the interviews, desire the most agile factors in the organization. This could be, because they are the smallest organization. Where organization C is in size the smallest, they desire the most agile factors in their organization. The second smallest organization also has the second largest amount of desired agile factors. The largest organization has the most desired traditional factors. This shows that organization B desires traditional factors, which indicates that they have the need for more governance and control. The amount of IT and agile teams are respective to the size of the organizations.

Figure 24 visualizes the desired agile factors by organizational size. The desired factors are per organization from the factors table data. The order of the graph is based on the organization with as detail the organizational size.

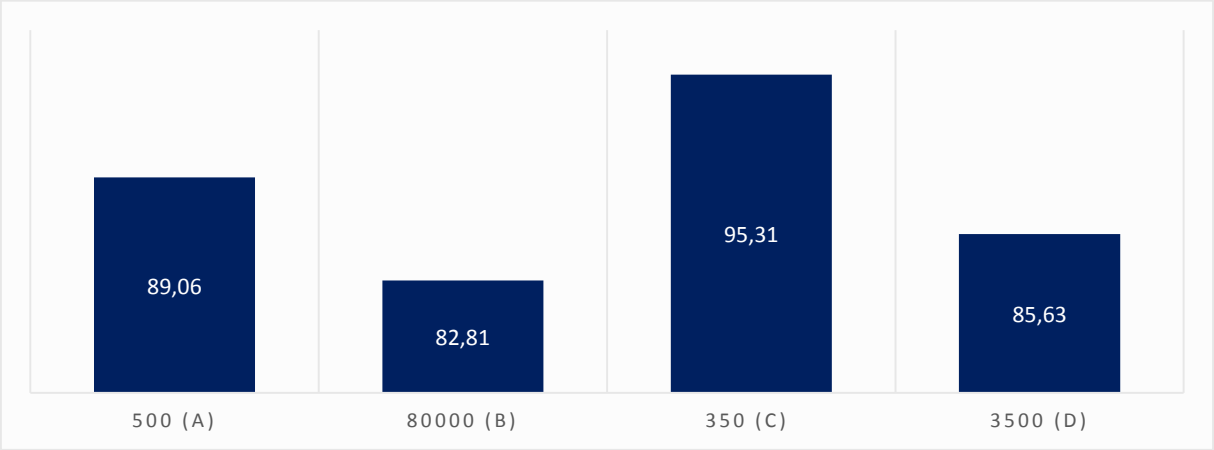


Figure 24: Desired factors by organization size

5.2 Categories

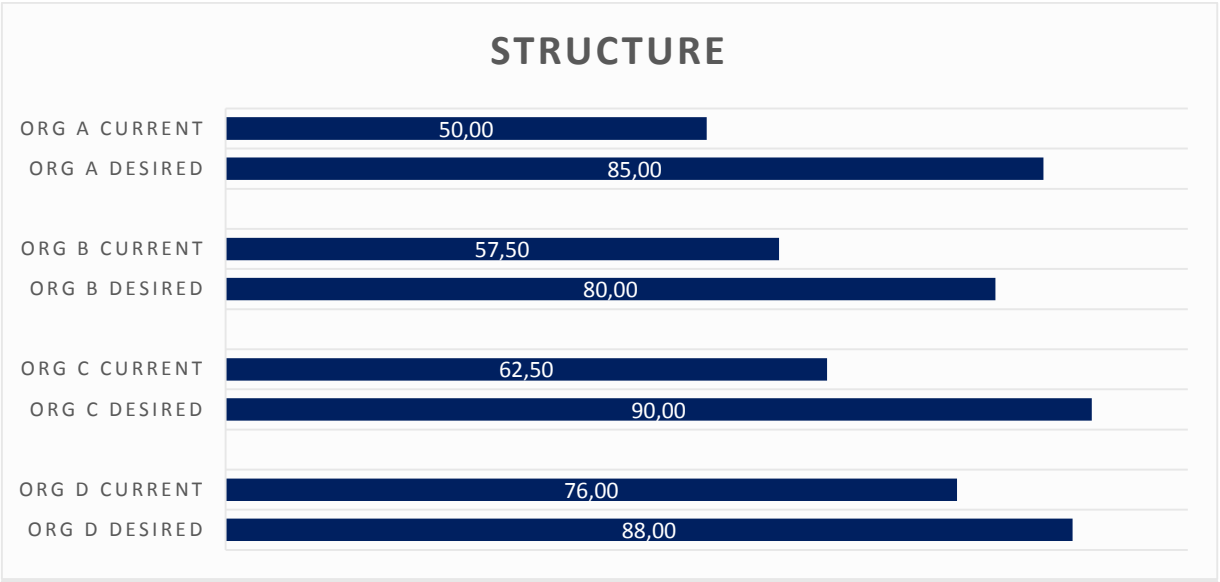
The factors table that we used in our interview was based on the four categories which were identified by McLoad and Macdonell (2011), which we interpreted as: Organizational Structure, Development Process, Project Content and Culture. When we combine the results of all the organizations, we can see in which category, the organization are strongest with agile factors and in which categories traditional factors are still present.

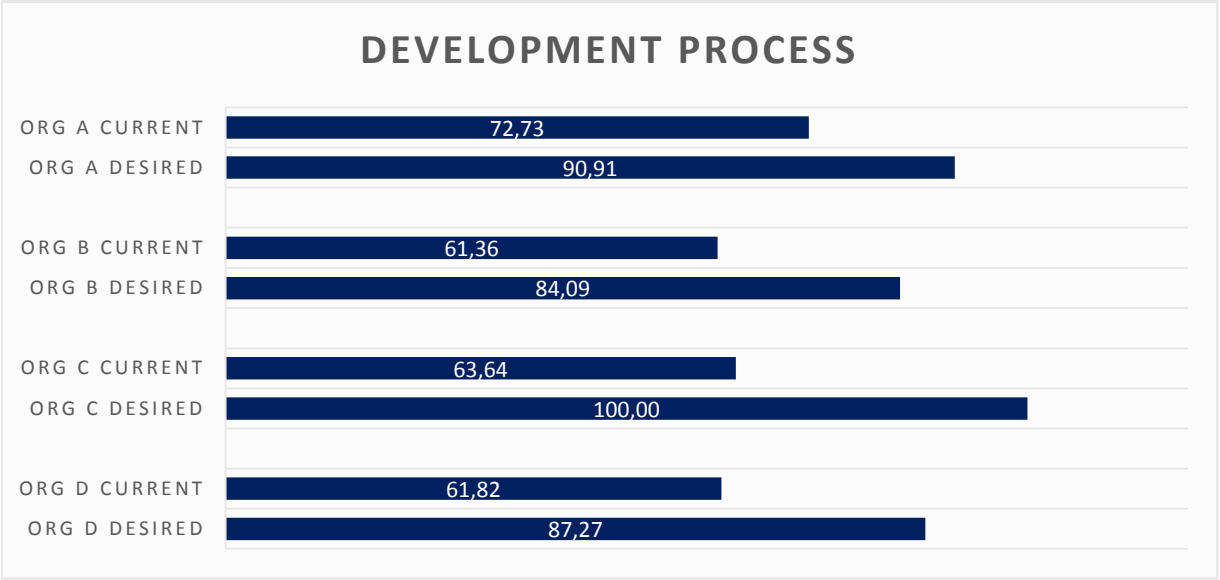
The graph below, shows that not all organizations are equally in the transition of their agile adoption. Organization A is leading the group when it comes to the Development Process and the Project Content. However, organization A is falling behind when it comes to Cultural factors. This would indicate that, despite the introduction of agile factors, the organization is still experiencing an traditional culture. A more traditional culture comes with more governance and regulations.

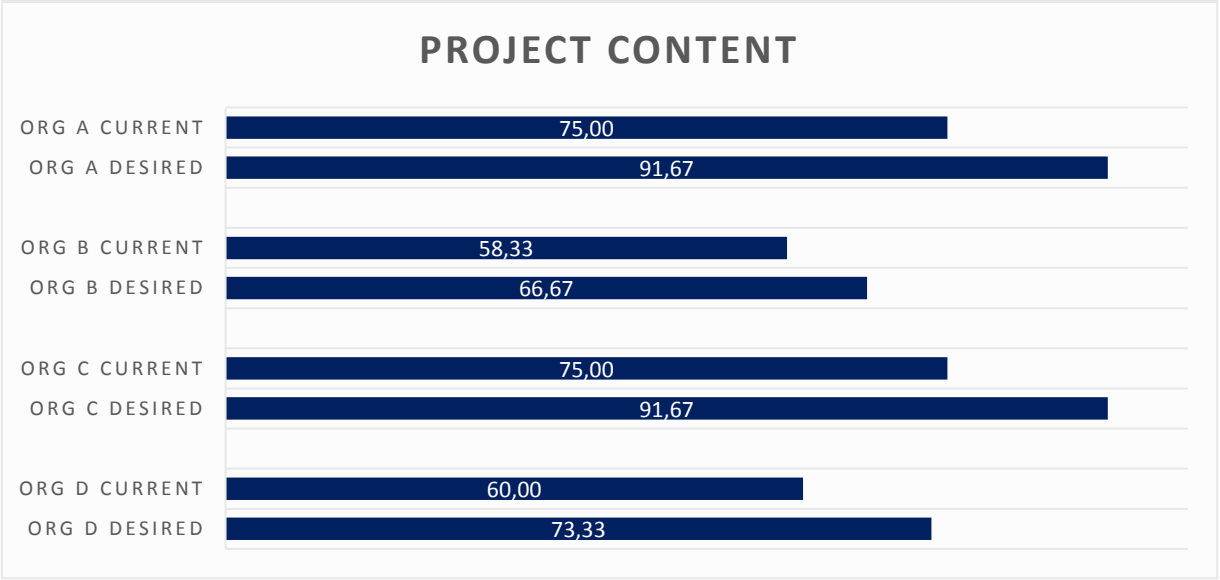
When we look at the desired culture for organization A, we see that the desire for a more agile culture is present. The explanation for this might be that the transition to agile was initiated as a top-down approach. In this organization, the transition started at the IT development department, and is slowly extending to the rest of the organization. in our interviews we discussed that the directors of this organization are very traditional and lead in a command and control style of leadership. This can explain the low amount of agile factors in culture.

In organization B the culture is perceived to be more agile than in the other organizations. As culture is one of the most difficult aspects to change in an organization, this deviation can be explained by the amount of time, spend on the agile transition so far. Organization B spend 4,5 years on the agile transition.

In organization D the culture is also perceived as more agile, in comparison of the other two organizations. What both of these organizations have in common, is that both organizations have a Top-Down approach of the agile transition, where management is encouraging the agile transition. However, organization D has only started the transition one year ago.


STRUCTURE
ORG A CURRENT
50,00
ORG A DESIRED
85,00
ORG B CURRENT
57,50
ORG B DESIRED
80,00
ORG C CURRENT
62,50
ORG C DESIRED
90,00
ORG D CURRENT
76,00
ORG D DESIRED
88,00


DEVELOPMENT PROCESS
ORG A CURRENT
72,73
ORG A DESIRED
90,91
ORG B CURRENT
61,36
ORG B DESIRED
84,09
ORG C CURRENT
63,64
ORG C DESIRED
100,00
ORG D CURRENT
61,82
ORG D DESIRED
87,27


PROJECT CONTENT
ORG A CURRENT
75,00
ORG A DESIRED
91,67
ORG B CURRENT
58,33
ORG B DESIRED
66,67
ORG C CURRENT
75,00
ORG C DESIRED
91,67
ORG D CURRENT
60,00
ORG D DESIRED
73,33

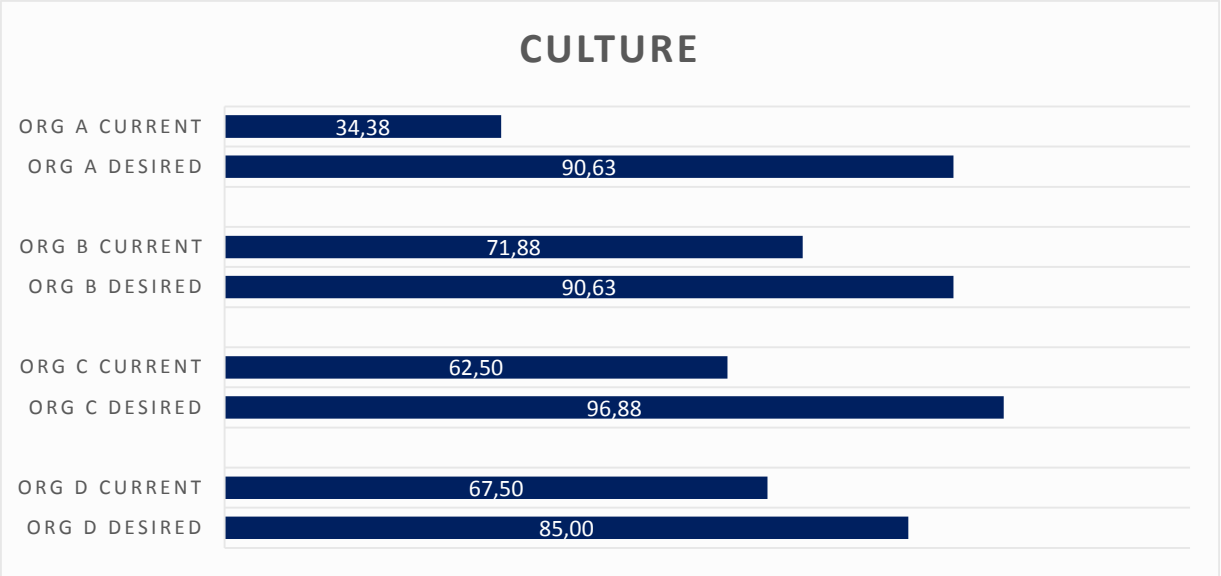


Figure 25: Current and desired results from the factors table, split per organization and divided per category.

5.3 Recommendation points for organizations

5.3.1 Factors Table

In our literature study we identified four categories, which are important in the adaptation of new software development methods. Within these categories we identified factors, based on literature study, which can be used identity what the perception of an employee is, regarding that factor. For each factor we identified an agile form and a traditional form, where the agile form is based on the available agile studies and the traditional forms relate more to traditional governance and regulated organizations.

We placed these factors in a table, which we used during our interview to determine what their perceptions where of the organization. We did this to test multiple aspects of the organization, in relation to the agile transition where the organizations are in.

This table can be read as followed. There are four categories, each with a different set of factors. The factors are aligned with both an agile factor displayed on the left side and the traditional factor on the right side. Below the factors is a description, which sets the context of the two factors.

The same table, with the same factors are used during each interview. Meaning that each person, despite the role the person has in the organization, has filled out the same table. This gives us inside in how different roles in the organization perceive the same factors.

Below we display the table, which factors are used during our interview. The actual table which is used during our interviews is attached in appendix 4. In this table the factors were shuffled from sides to eliminate the change that persons would just choose agile factors based on the fact that they were all easily identified.

Category	Structure	
Factors	Dedicated teams	People are part of multiple teams and projects
Description	In dedicated teams, the people are in a fixed team. On the other side, there are people whom are part of multiple teams and projects at the same time.	
Factors	More cross functional teams	Specialist line organization
Description	A cross-functional team is a group of people with different functional expertise working toward a common goal. E.g. finance, marketing and operations. This can include people from anywhere in the organization. Where a specialist line is a group or department with specialized people with the same skill set.	
Factors	More flexible governing rules	Management dictates rules
Description	Teams can adapt the way they are working, by simple governing rules, set by the organization, while in traditional governance the way of working is directed from top down.	
Factors	Decentralized decision-making	Centralized control
Description	Decisions are made in meeting with the right people, with subject experts and on the spot. Degree of decisions autonomy for teams is higher. Centralized control is a more traditional governing structure where the managers give top-down decisions.	
Factors	Informal communication	Formal communication
Description	In traditional structures, there is a degree of formality, and people are only to address their direct superiors. Contact goes to through official channels. Where in agile based organizations, communication is much more informal.	
Factors	Shared workspace	Separated workspace
Description	The development team has a dedicated place where they operate, where separated workspaces, everyone is just working where they are or where they like and are in contact with phone or instant messaging.	
Factors	Co-located developers	Developers located anywhere
Description	Co-located developers, means that all developers are on site, not necessarily sitting together, but are within reach. Developers located anywhere, meaning that your team could be scattered around the world and is communication electronically.	
Factors	Small team size	Large team size
Description	A small team is considered from 5 – 9 people with different roles. Whereas an large team exceeds this number.	
Factors	Extended support staff (also knowledge exchange)	Basic support staff (administration, HR, finance, safety)
Description	The supporting staff in an organization supplies besides the basic administration also additional services and coaching to the operational core. The supporting staff houses a center of experts whom support operations.	
Factors	Focus on updating systems	Focus on maintenances (legacy systems)
Description	The organization focusses on keeping systems up to date by new development. Whereas focus on maintenance results in a situation where the existing software is just maintained.	

Category	Development Process	
Factors	Iterative process	Linear process
Description	In an iterative process the team undergoes the same cycle of the project every sprint (2-3 weeks) until the work is complete. Whereas in a linear process, the work is created in one process and benefits and delivering take place at the end of the project.	
Factors	Value driven - Adaptive approach	Plan-driven - Predictable approach
Description	A value driven approach means making decisions or altering decisions if it benefits the value of the product and the process. Plan driven is a more predictable approach where the plan is executed.	
Factors	Incremental delivery (continuous)	All in once delivery
Description	Incremental delivery means that the delivered software grows with each increment. Users can get to know the software feature for feature whereas with all in once delivery, the entire program is delivered all in once.	
Factors	Direct Value	Future Value
Description	Direct value sacrifices documentation and risk analysis, to supply direct value by faster creating working products. Future value has more documentation and risk analysis, but the value for the business, is created in a later stage.	
Factors	Test every Iteration	Test at project completion
Description	Testing every iteration results in issues getting noticed much earlier, when the knowledge of the development is still present in the teams. This results in slower development. Testing when all the code is done can result in problems, from which the origin is unknown, and the bug needs to be find in far more code.	
Factors	Interactive input from customer - High client involvement	Predefined user requirements - Low client involvement
Description	During development iterations, the customers sees the software being created, the customer can speak-up for additional input, so the software is created according to up-to-date input from the customer. With predefined user requirements, the user sees the software when it's done, the result will be, what was specified at the beginning of the project.	
Factors	Low refactoring cost	High refactoring cost
Description	Because the customer is involved at each iteration of the project, changes can be accommodated in an much earlier stage. This has low cost involvement. However, if the product is finished and changes need to be made, the supplier needs to start their development team up again to change something, which could be embedded in the program already.	
Factors	Allow change in scope	Deliver Fixed Scope
Description	Because the customer is involved in the project, the scope of the project can change, due to the fact that the customers realizes that the software in development, will only fix some of their issues, and not all. This choice can translate into a longer project, but with more value to the customer.	
Factors	Teams are integral part of estimation	Management provides estimates
Description	When keeping the team as part of the estimations of progress and planning, a more accurate estimation can be produced, due to the fact that the team is closer to the actual product. Management however, have more insight in future planning and know how long the estimates should be to meet deadlines.	

Factors	Continuous improvement each iteration cycle	Lessons learned improvement after project
Description	After each iteration, retrospectives can be used to improve the process during the project. Retrospectives can also take place at the end of a project, to improve the next project.	
Factors	Learning through coaching, workshops, games	Formal education only
Description	Through coaching, workshop and games, the organization can learn the teams new methods in an interactive and fun manor. Formal education however, teaches new methods in an more formal setting.	

Category	**Project Content**	
Factors	Small project size	Large project size
Description	Small project size means that projects are defined into smaller projects, to make them manageable and better plan able. When working with small teams, delivering small pieces of a smaller project, which in itself can bring value to the organization.	
Factors	Low amount of documentation	High amount of documentation
Description	Due to low documentation, the developers need to meet up to further define their work. With each other or with customers. With more documentation, all requirements are clear, but creating the documents takes more time.	
Factors	Customization and in-house development	Off-the shelf software
Description	Customization and in-house development is a way to personalize software within the organization. This can result in creating additional benefit to the software, this requires developing next to managing the software. Off-the shelf software does not require development and just management. However, the off-the shelf software may not supply all functions required from the software.	

Category	**Culture**	
Factors	Leadership and collaboration	Command and control
Description	The project team is being supported by certain roles in the organization, e.g. Project manager removes any barriers hindering the core Agile teams. Developers have suggestion they can put forward, due to that they are the experts. With command and control it is execute the work that is specified.	
Factors	Collaborative environment	Competitive environment
Description	Knowledge is shared due to collaboration to innovate work. Within teams the knowledge is shared and collaborative ways of working are used to gain the best results from anyone in the team. In an competitive environment, people can be triggered to perform at their best. Trying to think of new ways to improve and outsmart competitors.	

Factors	Management commitment and collaboration	Management as controller, not involved in content
Description	Management commitment and collaboration results in the involvement of management with the development teams. Whereas management as an controller, the management control the development team from an managerial position, management is not involved in with the development team and the gap in content between the teams and management is larger.	
Factors	Management is encouraging improvements	Management is dictating ways of working
Description	Management is encouraging improvements made by the development team. This means that the development teams has the autonomy to improve their ways of working by themselves. Whereas management is dictating the ways of working, the teams does not have the autonomy to decide their own ways of working and governed by the methods set by the management.	
Factors	Transparency of goals, resources and progress	Unclear goals, resources and progress
Description	Transparency of goals, resources and progress means that everyone related to the project in some manner should be able to know or at least access these items. If this is not the case, if a business member cannot access this data, for the business, it's unclear. Unclear goals, resources and progress can also be related to management keeping this data unclear as a means to control information.	
Factors	Strategy and vision is embraced by the entire organization	Strategy and vision is something for top-managers
Description	Strategy and vision is something that is decided at top-management level and is used by them to guide the organization accordingly. Managers and operations know little about the strategy and vision of the organization and don't have to. Strategy and vision are considered to be something that top-management should be bothered with.	
Factors	Decisions are made based on metrics	Decisions are based on opinions
Description	Decisions are made based on metrics, meaning that the management is using metrics to make decisions. Whereas decisions are based on opinions.	
Factors	Boards and comities act as teams	Boards and comities are political playgrounds
Description	Boards and comities act as teams, they work together to strive to create the largest value for the organization. Boards and comities can also become political playgrounds where politics dictate the preferences and the outcome.	

5.3.2 Top 5 gap questions

With 10 interviewees mentioning that this factor need to be adopted in an agile way, the most mentioned factor were;

Boards and comities act as teams	Boards and comities are political playgrounds
Boards and comities act as teams, they work together to strive to create the largest value for the organization. Boards and comities can also become political playgrounds where politics dictate the preferences and the outcome.	

The next for factors were mentioned by 8 interviewees over the different organizations;

More flexible governing rules	Management dictates rules
Teams can adapt the way they are working, by simple governing rules, set by the organization, while in traditional governance the way of working is directed from top down.	

Low refactoring cost	High refactoring cost
Because the customer is involved at each iteration of the project, changes can be accommodated in an much earlier stage. This has low cost involvement. However, if the product is finished and changes need to be made, the supplier needs to start their development team up again to change something, which could be embedded in the program already.	

Allow change in scope	Deliver Fixed Scope
Because the customer is involved in the project, the scope of the project can change, due to the fact that the customers realizes that the software in development, will only fix some of their issues, and not all. This choice can translate into a longer project, but with more value to the customer.	

Learning through coaching, workshops, games	Formal education only
Through coaching, workshop and games, the organization can learn the teams new methods in an interactive and fun manor. Formal education however, teaches new methods in an more formal setting.	

These were the factors which were mentioned most during our interviews as factors which would be desired to change towards more agile. However, this is the overall average of the four organizations. It could be possible that some organizations, don't relate to one or more of these factors. The next page shows the distribution of the points per question and per organization.

Agile aspects	Traditional aspects	Q#	Category	A	B	C	D
Dedicated teams	People are part of multiple teams and projects	1	Structure (organizational)	0	0	0	0
More cross functional teams	Specialist line organization	2		2	2	1	0
More flexible governing rules	Management dictates rules	3		3	2	1	2
Decentralized decision-making	Centralized control	4		1	0	2	0
Informal communication	Formal communication	5		1	1	1	1
Shared workspace	Separated workspace	6		1	0	1	2
Co-located developers	Developers located anywhere	7		1	2	0	1
Small team size	Large team size	8		1	0	2	0
Extended support staff (also knowledge exchange)	Basic support staff (administration, HR, finance, safety)	9		3	1	1	0
Focus on updating systems	Focus on maintenances (legacy systems)	10		1	1	2	0
Iterative process	Linear process	11	Development Process	1	1	0	0
Value driven - Adaptive approach	Plan-driven - Predictable approach	12		2	1	2	1
Incremental delivery (continuous)	All in once delivery	13		0	-1	1	1
Direct Value	Future Value	14		-2	2	2	2
Test every Iteration	Test at project completion	15		0	0	1	1
Interactive input from customer - High client involvement	Predefined user requirements - Low client involvement	16		2	1	2	0
Low refactoring cost	High refactoring cost	17		1	3	2	2
Allow change in scope	Deliver Fixed Scope	18		2	1	3	2
Teams are integral part of estimation	Management provides estimates	19		0	0	1	2
Continuous improvement each iteration cycle	Lessons learned improvement after project	20		0	1	0	0
Learning through coaching, workshops, games	Formal education only	21		2	1	2	3
Small project size	Large project size	22	Project	2	2	3	0
Low amount of documentation	High amount of documentation	23		1	0	0	2
Customization and in-house development	Off-the shelf software	24		-1	-1	-1	0
Leadership and collaboration	Command and control	25	Culture	1	0	1	2
Collaborative environment	Competitive environment	26		0	0	1	0
Management commitment and collaboration	Management as controller, not involved in content	27		3	0	3	1
Management is encouraging improvements	Management is dictating ways of working	28		2	0	2	1
Transparency of goals, resources and progress	Unclear goals, resources and progress	29		2	0	1	0
Strategy and vision is embraced by the entire organization	Strategy and vision is something for top-managers	30		3	1	2	0
Decisions are made based on metrics	Decisions are based on opinions	31		3	1	0	2
Boards and comities act as teams	Boards and comities are political playgrounds	32		4	4	1	1

Figure 26: Table data per category and per organization

5.3.3 Top gaps per organization

For each organization, the identified gap questions are different, though all the top 5 overall gap questions are presented in these organizations. Below is an overview of the top gaps per organization. These tables show the agile factor on the left and the traditional factor on the right. In each of these cases, the agile factor is more desired than the traditional factors.

5.3.3.1 Organization A

Boards and comities act as teams	Boards and comities are political playgrounds
Boards and comities act as teams, they work together to strive to create the largest value for the organization. Boards and comities can also become political playgrounds where politics dictate the preferences and the outcome.	

More flexible governing rules	Management dictates rules
Teams can adapt the way they are working, by simple governing rules, set by the organization, while in traditional governance the way of working is directed from top down.	

Extended support staff (also knowledge exchange)	Basic support staff (administration, HR, finance, safety)
The supporting staff in an organization supplies besides the basic administration also additional services and coaching to the operational core. The supporting staff houses a center of experts whom support operations.	

Management commitment and collaboration	Management as controller, not involved in content
Management commitment and collaboration results in the involvement of management with the development teams. Whereas management as an controller, the management control the development team from an managerial position, management is not involved in with the development team and the gap in content between the teams and management is larger.	

Strategy and vision is something for top-managers	Strategy and vision is embraced by the entire organization
Strategy and vision is something that is decided at top-management level and is used by them to guide the organization accordingly. Managers and operations know little about the strategy and vision of the organization and don't have to. Strategy and vision are considered to be something that that top-management should be bothered with.	

Decisions are made based on metrics	Decisions are based on opinions
Decisions are made based on metrics, meaning that the management is using metrics to make decisions. Whereas decisions are based on opinions.	

5.3.3.2 Organization B

Boards and comities act as teams	Boards and comities are political playgrounds
Boards and comities act as teams, they work together to strive to create the largest value for the organization. Boards and comities can also become political playgrounds where politics dictate the preferences and the outcome.	

Low refactoring cost	High refactoring cost
Because the customer is involved at each iteration of the project, changes can be accommodated in an much earlier stage. This has low cost involvement. However, if the product is finished and changes need to be made, the supplier needs to start their development team up again to change something, which could be embedded in the program already.	

5.3.3.3 Organization C

Allow change in scope	Deliver Fixed Scope
Because the customer is involved in the project, the scope of the project can change, due to the fact that the customers realizes that the software in development, will only fix some of their issues, and not all. This choice can translate into a longer project, but with more value to the customer.	

Small project size	Large project size
Small project size means that projects are defined into smaller projects, to make them manageable and better plan able. When working with small teams, delivering small pieces of a smaller project, which in itself can bring value to the organization.	

Management commitment and collaboration	Management as controller, not involved in content
Management commitment and collaboration results in the involvement of management with the development teams. Whereas management as an controller, the management control the development team from an managerial position, management is not involved in with the development team and the gap in content between the teams and management is larger.	

5.3.3.4 Organization D

Learning through coaching, workshops, games	Formal education only
Through coaching, workshop and games, the organization can learn the teams new methods in an interactive and fun manor. Formal education however, teaches new methods in an more formal setting.	

5.4 Agile Coaches

Agile coaches were interviewed in each of the organizations. Between these agile coaches and the agile coaches which were encountered during this study, different perceptions were identified. Some agile coaches had the perception that agile is a tool which must not be changed or adapted to an organization, but be implemented in the predefined form. Other agile coaches had the perception that agile supplies methods and tools which could help your organizational processes by supplying a supportive method and framework. In out interview, we encountered both these types of agile coaches. While their desired outcome differs, their current perspective of the, regarding agile factors, is relatively low.

Figure 27 visualizes the difference between the average of the organizations against the perception of the agile coaches in the current situation. When comparing the agile desire score, with that of the rest of the organization, something stands out. Agile coaches do not desire all agile factors.

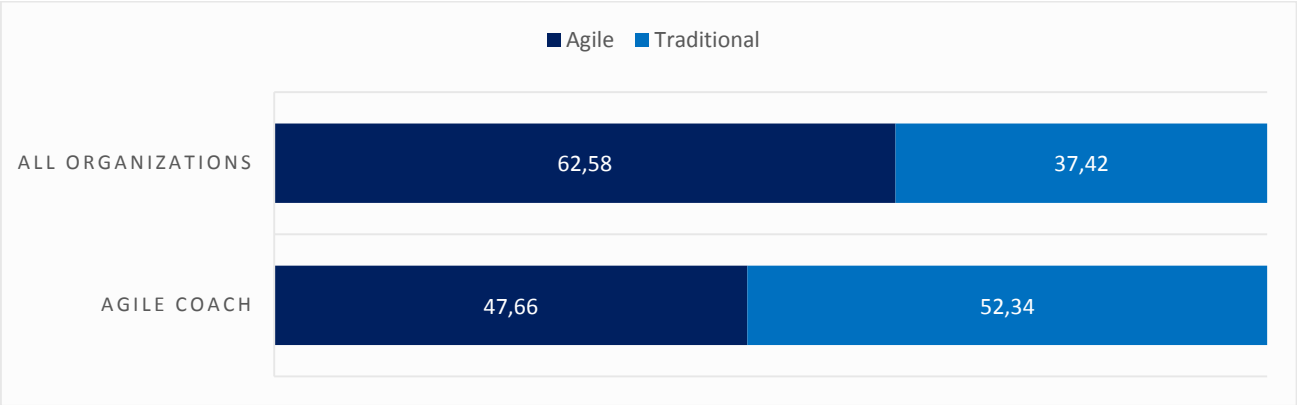


Figure 27: Difference between the organizations and the agile coaches

5.4.1 Internal vs external

During our study we encountered both internal and external agile coaches in companies. The internal agile coaches seem to be more in touch with preserving the internal organization. External agile coaches are more rigorous in the adoption methods. This was observed during our time spent in the different organizations and also confirmed by an agile coach who was hired by one of the organizations. This is an agile coach who is working with agile coaches from different external organizations. The external coaches bring their own methods and receive little regulations or guidance regarding work methods. This means that different teams, which each have their own agile coach, is guided in a different manner and are being guided differently during agile rituals. This results in an work setting where team don't operate within the same context.

5.4.2 Individual vs corporate consultancy.

Another result that we encountered during our time spent at the organizations was that individual agile coaches or self-employed agile coaches have a different view on agile and transition methods than agile coaches from corporate consultancy firms. During our research we confirmed this statement from both individual and corporate coaches.

We observed that the individual coach have a different approach to the agile transition. The individual coach seems to be more involved with the organization and is supporting this organization with practices. The coach adapt this practices accordingly. Corporate coaches seem to be working more with standards and "by the book" practices. This is confirmed by an agile coach who said, "agile is something which should not be changed, why change what is proven to work".

5.4.3 Are coaches eager to learn?

As part of our interview, we asked “how would you rate your agile knowledge on a scale from 1 to 10”. Where 1 would be the lowest score and 10 the highest. If someone would answer with a 10, is would mean that the person knows everything there is to know on the subject and that there is nothing left to learn. Is this the case with agile software development? Would it be possible for one person to know everything about this subject? A lot of conferences and gatherings are organized to share knowledge were people learn more about agile each time. Is it realistic to mention that your knowledge of agile is a 10? Or even a high grade? Since agile is an upcoming and trending topic in larger organizations these days.

5.4.4 Guidance in an organization

In an organization, agile coaches should be willing to take everyone, including management, by the hand and show them the way. Since agile is new to the managers, they should be coached as well. Some managers are self-learning or have previous experience with agile. They should work together with the coaches to embed agile in the organization. in our case organizations, we see an difference in approach. Where some organizations the managers and the agile coaches work together and some where the management and the agile coaches are not involved. in this case, some of the management is actually resisting the agile change. Is this justified?

If an manager feels that agile is not beneficial for the processes or the organization itself, should he keep quiet? Agile coaches need the support of the management to create an environment where the organization can embrace agile. If management is resisting the agile coaches or the agile transition, there is a feeling of distrust. Is the agile coach really in line with the organization and is he implementing agile to benefit the organization or is the coach too indoctrinated and is he coping examples of other implementations and is he “doing it by the book”?

In some organizations we heard the quote ”agile is something which should not be changed, why change what is proven to work”. This would mean that the coach is not paying a lot of attention to the organization or is trying to change the organization to fit the agile context. For larger and more regulated organizations, the agile context or sweet spot, which was identified by Kruchten (2003), would not be possible to achieve. Keeping this in mind, would these agile coaches bring benefit to the organizations?

5.5 Observations

In the four organizations, different observations were done on-site. During the observations we encountered multiple rituals per organization. We tried to attend the same rituals in each of the different organizations. In all four of our organizations we attended a ritual which we perceived as a Sprint Planning meeting. Each of these rituals are executed slightly different. Where on organization is executing the rituals formally, other organizations are more casual in the formalization of the rituals.

We see this factor also in the culture. If rituals are executed in a casual and nonchalant manner, the culture of the organization (in the case of an large organization it is more the culture of the column or department) reflects the behavior which was observed in the rituals.

Each organization is doing the rituals slightly different. This is because each organization works in a different way and have their own processes to which agile is complementary. The amount of differentiation in the rituals within the same company, is due to the amount and the source of the agile coaches. With the source of the agile coaches we mean from what company are the agile coaches, are they internal or external coaches. If they are external, are they corporate or self-employed? This matters because as discussed earlier, these coaches have their own “style” of guiding rituals. If the organizations have multiple rituals happening at once, there were multiple coaches present, spread over all the rituals. If the coaches came from a different source, the guidance of the rituals differs. This was also observed at one of the organizations.

What we take away from the time spent at the rituals is that Culture and Guidance are the two strongest factors which influence the rituals. The actual process of the rituals is based on the organizational processes and will therefor always be slightly different.

5.6 Contextually of Regulations

Regulation exist in each of the organizations. Even though some roles in the organization would argue that the regulations are a necessity, regulations can hinder the organization in their processes and work. Everyone who we have interviewed had something to mention on hindering regulations, as discussed in Figure 28.

The question for each organization is, what are the right regulations to have and what could be regulated less. This is a hard question to answer, because differ roles have different perspectives about these regulations.

In our study we found that in the organizations, regulations are in place which hinders the software development methods from creating benefit for the organizations. Regulations should be in place to support and perhaps protect the operations of the organization. If one of these operations is software development, it should not be hindered by these regulations.

Each organizations have their reasons for having these regulations in place. In one case, the regulations are in place because of the parent organization is demanding this. In other cases the regulations are in place because the transition towards agile software development methods is going slower than expected. Whatever the reason, the different persons we interviewed in our research, have experienced hinder from regulations. Below we created an table which identifies what regulations were perceived in each organization, divided over the organizational roles.

The regulations in this figure 28 are the regulations which are perceived to hinder the organization in its development process. The source for this data is the interview data.

Regulations				
	Directors/Management	**Agile Coaches**	**Business**	**Operational Staff**
Org A	Software	Traditional management Matrix organizations	Traditional management Matrix organizations	Too much centralization
Org B	Lack of governance Guidance of explicit roles	Procedures Infrastructure	Too much regulations Governed by the governing entity (State)	Lack of stability
Org C	Procedures	Matrix organizations	Multiple methods Politics and boards Traditional organization Communication	Customer involvement Communication
Org D	Traditional management Regulating procedures	Regulating administrative procedures	Communication Matrix Organizations	Infrastructure

Figure 28: Perceived regulations across organizational roles.

6 CONCLUSIONS

In the conclusions section, we discuss the conclusions that we can make from our research. Based on the conducted research, consisting of spending 10 months on-site at 4 large regulated organization, doing 16 observations (1490 minutes of formal observations) and spending countless hours of informal observing organizational processes and doing 17 interviews (about 1020 minutes of interviews) across multiple organizational levels, we conclude that: (1) Due to organizational context, the balance between the organizational environment and the software development method is different for each organizations. (2) The agile factors and regulations in the organizations are differently perceived by the multiple organizational levels.

6.1 Answering the research questions

6.1.1 Sub Question 1

- *What organizational factors, in large software developing organizations, are different when looking at agile and traditional perceptions?*

To answer this sub question, we did a literature study on the topic of regulated factors and the contextualization of agile. This research resulted in a table which we used in our interviews to gather data from different persons spread over different organizations. The resulting table consists of agile factors against traditional factors. These factors are divided in 4 different categories; Structure, Development Process, Project content and Culture. The First 10 factors are organizational Structure factors, followed by 11 Development Process factors. 3 Project content factors and 8 Culture factors. This table is located in paragraph 5.3.1.

6.1.2 Sub Question 2

- *What are the organizational factors, within large regulated organizations, which are perceived to stand out?*

In order to determine which factors stand out, we looked at what was perceived by the interviewees per organization. We calculated the difference between what was perceived and what is desired. Due to our table, we could measure this with numbers.

When we break down the results, we determined that the largest gap in the current and desired factors are located in the culture category.

Organizations	A	B	C	D	Total	# Questions	Representation	% out of 100
Structure	14	9	11	6	40	10	4,00	25,08
Development Process	8	10	16	14	48	11	4,36	27,36
Project content	2	1	2	2	7	3	2,33	14,63
Culture	18	6	11	7	42	8	5,25	32,92

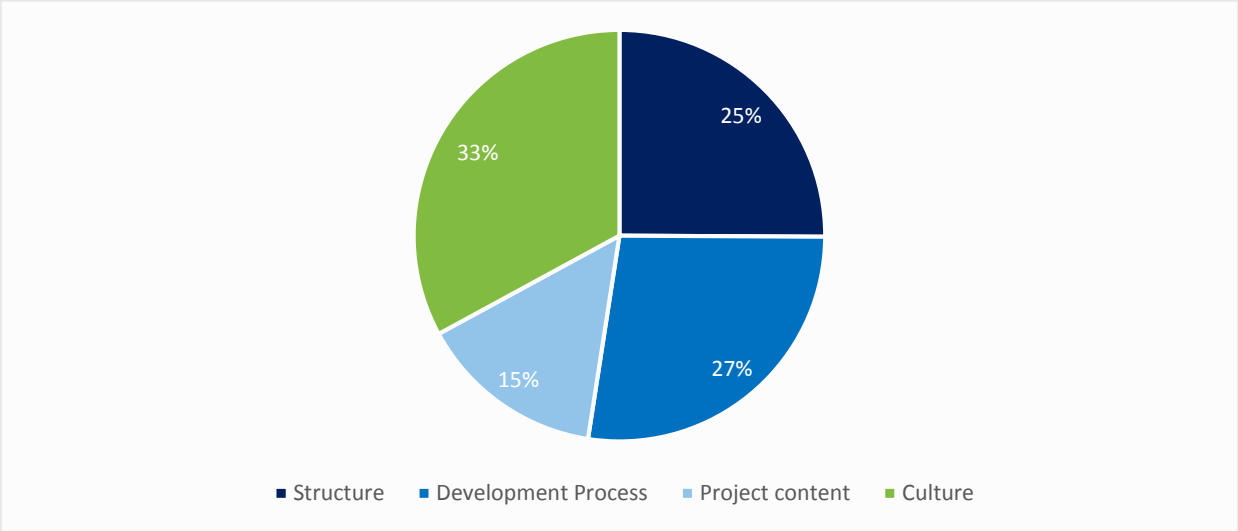


Figure 29: Average of deviation in factors

Figure 29 shows us that relatively, 33% of the difference in factors, can be found in the culture category. This means that within this category, the interviewees had the most desire to adopt more agile factors in the organization. When looking at our organizations, which are large and regulated, this is to be expected.

The second largest percentage is found in the Development Process category. Agile methods, are in origin software development methods, such as Scrum. All of our researched organizations are trying to implement such an agile method. Difficulties here lies in the shift from traditional toward agile. The agile practices and processes need to be in line with the organization, which takes time.

6.1.3 Research question

- *How to achieve the right balance of adapting the organizational environment on one hand and adapting the software development method on the other, in order to create the biggest benefit for the organization?*

The answer to this question is very contextual. The balance between the organizational environment and the software development method would be different for each organization, since the organizational context is different. To address this question, we have developed the factors table presented in chapter 5.3.1.

Both the organizational environment and the software development method, determine if there is a balance, within the organizational context. For some organizations the emphasis in the ratio of organizational environment and software development method is more focused on the organizational environment, which would indicate that more regulations has to be in place, in order to stay compliant. This means that the agile factors we identified in our factors table, can only be present if they are in compliance with the regulations set by the organization. If this is the case, a balance is present between the organizational environment and the software development method.

Even though the balance point is not the same for each organization, it is possible to better understand where the balance point for each organization is. Using our factors table, the current state of agile factors in the organization can be identified, in relation to traditional factors. Based on the perceived desired outcome of the factors table, the benchmark has been set to further invest in agile factors in the organization. Because the desired state of the organization consists of more agile factors in the organization, we conclude that the organizations still can invest in agile factors if the compliance regulations are identified in the organization.

6.2 Validity considerations

To validate our research, we used the methods of Robert K. Yin (2013).

6.2.1 Construct validity

We identified the most fitting method of data collection for this study. The semi-structured interviews combined with the factors table enabled us to get the desired data from the desired sources. We got the opportunity to select the interviewee's, which enables us to make sure that the correct roles in the organization were represented. The observations conducted in the organizations complimented the interview data.

6.2.2 Internal validity

In our results and discussion section we argue, based on the gathered data, using graphs and charts. These are visual representations of the data. Because the factors table had a Boolean outcome, the data is comparable. We were able to identify where the largest gap is located in the factors. We mention in our document where the data is coming from and how we get to the displayed results.

6.2.3 External validity

We used the same methods of data gathering at each of the organizations. With each interviewee we used the same interview guide and the factors table. The same roles were selected in all the organizations. Because the organizational context is different in each organization, the data collected would show different outcomes, even though the methods used are the same. This results in yet more arguments that agile benefits are contextual.

6.2.4 Reliability

During our data collection we used protocols and guides to ensure that each data source is collected in the same manner. We created an interview guide to standardize most of the interview questions. The factors table did not change during our data collection, so each person filled in the same table.

6.3 Recommendations for practice

From our study we have some recommendations which will benefit every organization.

- Have a meeting with the management, directors and agile coach in your company. See if agile coaches share the same vision, mission and strategy for the organization. Make sure that the result benefits the organization. define a standard as a ways of working, so all employees are guided in agile aspects the same way.

- Measurements is something which the organizations in our research are not doing enough or not at all. The organizations are doing an agile transition, meaning that they are investing a lot of time and money in the transition, without measuring if the net method is actually beneficial for the organization. This concern was raised on multiple occasions in our findings.

- Communication is always important, in each organization. During our interviews we encountered interviewees who have no idea how and why agile in implemented in the organization.

Even though the balance differs in each organization, we conclude that the following aspects need to be in place and understood thoroughly, in order to gain benefit of the software development method.

6.3.1 Role of Agile Coaches

The agile coach is present in the organization for knowledge exchange and guidance. It is important that the organization selects the right agile coach which fits the organization. We concluded that there are different types of agile coaches available, which have different believes and styles. Each of these coaches will achieve different results in the organization.

Types of agile coaches:

- Internal
- External self-employed
- External Corporate

For an organization it would be difficult to identify which type of coach would best fit their organization. Since agile is a relative new topic, this research is not well known.

6.3.2 Role of Management

Agile transition required the commitment of the entire organization. This includes management and directors. They need to initiate the transition is the organization, together with help from people with knowledge of agile or agile transitions. The coaches are there to guide transitions, but the execution need to come from management.

Management and directors need to be aware of what is happening at each level of the organization. If this is not the case, the management can't make decisions in general. Same goes for the agile transition. Together with the coach, the management should identify the struggles and impedance and remove them in line of the organizational best interests. This does not necessarily have to be in line with agile, in some cases it is possible that agile aspects are not beneficial for the organization.

6.3.3 Understanding of organizational culture

Culture is the one factor which returns throughout our discussion chapter. From this we can conclude that culture is very important of the agile transition in organizations.

We see in our results section that culture is the factor which stand out the most, when it comes to deviation. People want a less traditional culture (more informal) which means more leaderships and more management involvement. Employees want to become more involved with the organization.

If the organization has a culture which fits a traditional organization, this would indicate that agile is harder to implement. When we look at small organizations, culture is often informal and the organizational hierarchy is flat. This setting is, as previously defined, is more ideal for the implementation of agile methods in comparison.

6.3.4 Software development method in the organization environment

The organizational environment is the context in which the organization operates. The environment should support the operations of the organization. If the organization is developing software, these operations should also be supported by the organizational environment. This means that the organization need to adapt its environment accordingly. Making room for the software development method to be imbedded in the organization. However, this also means that the software development method, should fit the organizational context (structure, process, content, culture) in the first place.

The first question the organization needs to ask is, can the current organizational environment support an agile software development method? The second question is: is it possible for an agile software development method to work in the current organizational environment?

Both questions will most likely be answered no by large regulated environments, which means that adaptation of both the organizational environment and the agile context is required.

The organizational environment support all the organizational processes, also the ones which does not involve software development. These processes should not be hindered due to the adaptation of the software development method adaptation. This would both create and destroy organizational benefit.

The same principle goes for the software development method. To what extend is it feasible to adapt the method to an organizational environment, without losing the benefits that the method brings? At what point, does the method stop being beneficial and start becoming hinder?

6.4 Future work

Due to time limitations and the nature of large regulated environments, relatively little data is gathered to really generalize the data, which is not possible at this stage. With more data, collected by a survey/questionnaire, more statistical analysis would have been possible. In our study we looked at one person per role in an organization. Multiple persons per role would give a more ground for generalization.

REFERENCES

Abrahamsson, P., Warsta, J., Siponen, M., & Ronkainen, J. (2003). New directions on agile methods: a comparative analysis. *Software Engineering*.

Ambler, S. W. (2009). Scaling agile software development through lean governance. *Proceedings of the 2009 ICSE Workshop on Software Development Governance*.

Ambler, S. W. (2009). The Agile Scaling Model (ASM): Adapting Agile Methods for Complex Environments. *IBM*.

Awad, M. A. (2005). A comparison between agile and traditional software development methodologies. *University of Western Australia*.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). *Manifesto for Agile Software Development*. Retrieved from Agile Manifesto: http://agilemanifesto.org

Boehm, B. (1988). A spiral model of software development and enhancement. *Computer* , 61-72.

Boehm, B., & Hansen, W. (2000). Spiral development: Experience, principles, and refinements. *CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST*.

Cao, L., & Ramesh, B. (2007). "Agile software development: Ad hoc practices or sound principles? *IT professional 9.2*, 41-47.

Cawley, O., Wang, X., & Richardson, I. (2010). Lean/Agile Software Development Methodologies In Regulated Environments – State of the art. *Lean Enterprise Software and Systems*, 31-36.

Cheng, T.-H., Jansen, S., & Remmers, M. (2009). Controlling and monitoring agile software development in three Dutch product software companies. *Proceedings of the 2009 ICSE Workshop on Software Development Governance*, 29 - 35.

CMS. (2008). Selecting a development approach. *United States Department of Health and Human Services*.

Fitzgerald, B., Stol, K.-J., O'Sullivan, R., & O'Brien, D. (2013). Scaling Agile Methods to Regulated Environments: An Industry Case Study. *Proceedings of the 2013 International Conference on Software Engineering*, 863-872.

Hajjdiab , H., & Taleb, A. (2011). Adopting Agile Software Development: Issues and Challenges. *International Journal of Managing Value and Supply Chains*, 1-10.

Hardy, G. (2006). Using IT governance and COBIT to deliver value with IT and respond to legal, regulatory and compliance challenges. *Information Security technical report 11.1*, 55 - 61.

Hass, K. B. (2007). The blending of traditional and agile project management. *PM world today*, 1-8.

Hoda, R., Kruchten, P., & Noble, J. (2010). Agility in context. *ACM Sigplan Notices. Vol. 45. No. 10. ACM*.

ISTQB. (n.d.). *What is RAD model- advantages, disadvantages and when to use it?* Retrieved from ISTBQ Exam Certification: http://istqbexamcertification.com/what-is-rad-model-advantages-disadvantages-and-when-to-use-it/

Kettunen, P., & Laanti, M. (2008). Combining agile software projects and large-scale organizational agility. *Software Process: Improvement and Practice*, 183-193.

Kruchten, P. (2013). Contextualizing agile software development. *Journal of Software: Evolution and Process* , 351 - 361.

Laanti, M., Salo, O., & Abrahamsson, P. (2011). Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *Information and Software Technology 53.3*, 276-290.

Larman, C., & Basili, V. (2003). Iterative and Incremental Development: A Brief History. *Computer*, 47-56.

Leffingwell, D. (2014). Agile Software Development with Verification and Validation in High Assurance and Regulated Environments. *A Rally Software Development Corporation Whitepaper*.

Leffingwell, D. (2014). *Scaled Agile Framework*. Retrieved from Scaled Agile Framework: http://scaledagileframework.com

Letho, I., & Rautiainen, K. (n.d.). Software development governance challenges of a middle-sized company in agile transition. *Proceedings of the 2009 ICSE Workshop on Software Development Governance*.

Martens, M. (2014). Management accounting & control: Agile Systeemontwikeling. *MCA*, 30-38.

McLeod, L., & MacDonell, S. G. (2011). Factors that affect software systems development project outcomes: A survey of research. *ACM Computing Surveys (CSUR) 43.4* , 24.

Meso, P., & Radhika, J. (2006). Agile software development: adaptive systems principles and best practices. *Information Systems Management* , 19-30.

Mintzberg, H. (1980). Structure in 5's: A Synthesis of the Research on Organization Design. *Management science 26.3*, 322 - 341.

Mintzberg, H. (2006). *Structures In Fives Designing Effective Organizations.* Pearson.

Moe, N., Dingsøyr, T., & Dybå, T. (2009). Overcoming bariers to self-management in software teams. *Software, IEEE 26.6*, 20-26.

Nerur, S., & Balijepally, V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM 50.3*, 79-83.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 72-78.

Petersen, K., Wohlin, C., & Baca, D. (2009). The Waterfall Model in Large-Scale. *Product-Focused Software Process Improvement*, 386-400.

ProjectManagement. (n.d.). *Process/Project RAD - RAD - Rapid Application Development Process*. Retrieved from ProjectManagement.com: http://www.projectmanagement.com/content/processes/11306.cfm

Royce, D. (1970). Managing the development of large software systems. *IEEE WESCON. Vol. 26*, 328-338.

Serguei , K. (2004). Comparing eXtreme Programming and Feature Driven Development in academic and regulated environments. *Software Architecture and Engineering*.

Sommerville, I., & Kurkovsky, S. (n.d.). *Rapid Software Devlopment.* Retrieved from Department od Computer Science - Central Connecticut State University: http://www.cs.ccsu.edu/~stan/classes/CS530/Slides/SE-17.pdf

Stettina, C. J., & Hörz, J. (2015). Agile portfolio management: An empirical perspective on the practice in use. *International Journal of Project Management 33.1*, 140-152.

Stoica , M., Mircea, M., & Ghilic-Micu, B. (2013). Software Development: Agile vs. Traditional. *Informatica Economica*, 64-76.

Sutherland, J., Viktotov, A., Blount, J., & Puntikov, N. (2007). Distributed scrum: Agile project management with outsourced development teams. *System Sciences*.

Thakur, D. (n.d.). *Rapid Application Development (RAD) Model and its Advantages and Disadvantages of RAD Model.* Retrieved from Computer Notes: http://ecomputernotes.com/software-engineering/rapid-application-development

Thomas, J., & Baker, S. (2008). Establishing an agile portfolio to align IT investments with business needs. *AGILE'08. Conference. IEEE*, 252-258.

Tumbas, P., & Matkovic, P. (2006). Agile vs. Traditional methodologies in developing Information Systems. *Management Information Systems*, 15-24.

Verma, J., Bansal, S., & Pandey, H. (2014). Develop Framework for Selecting Best Software Development Methodology. *International Journal of Scientific & Engineering Research* , 1067-1070.

Vo, H. (2007). Software development process. *Preuzeto* .

West, D., & Grant, T. (2010). Agile Development: Mainstream Adoptation Has Changed Agility. *Forrester*.

Yin, R. K. (2013). *Case study research: Design and methods.*

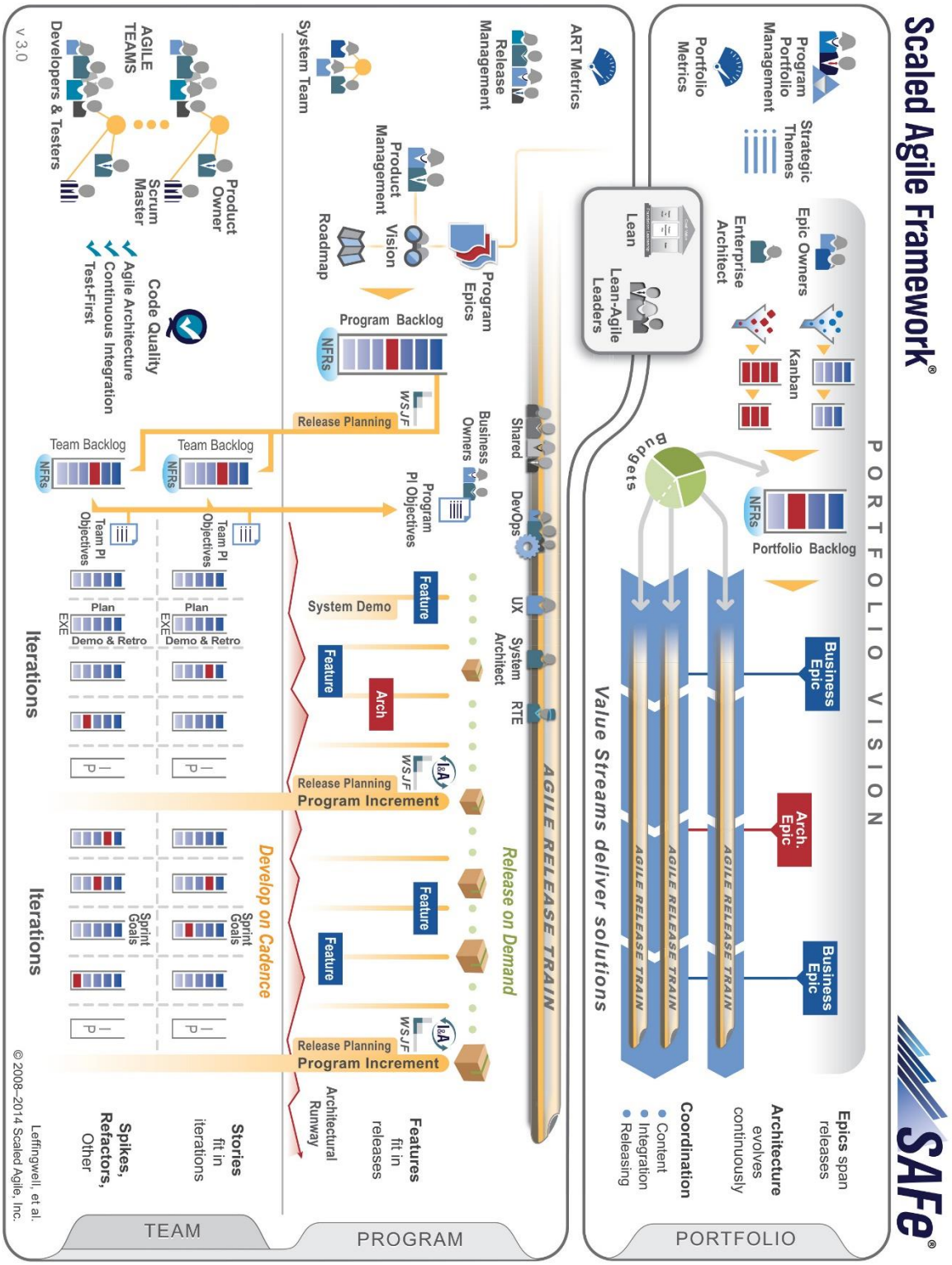
APPENDIX 1

Principles behind the Agile Manifesto (Beck, et al., 2001)

We follow these twelve 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 behavior accordingly.

APPENDIX 2


Scaled Agile Framework®
SAFe®
PORTFOLIO VISION
Program Portfolio Management
Strategic Themes
Portfolio Metrics
Epic Owners
Enterprise Architect
Kanban
Lean
Lean-Agile Leaders
Budgets
NFRs
Portfolio Backlog
Business Epic
Arch. Epic
Business Epic
AGILE RELEASE TRAIN
AGILE RELEASE TRAIN
AGILE RELEASE TRAIN
Value Streams deliver solutions
Epics span releases
Architecture evolves continuously
Coordination
Content
Integration
Releasing
PORTFOLIO
ART Metrics
Release Management
System Team
Product Management
Vision
Roadmap
Program Epics
Program Backlog
NFRs
WSJF
Release Planning
Shared
DevOps
UX
System Architect
RTE
AGILE RELEASE TRAIN
Business Owners
Program PI Objectives
Team Backlog
Team Backlog
NFRs
NFRs
Team PI Objectives
Team PI Objectives
Plan
Demo & Retro
Plan
Demo & Retro
EXE
EXE
Iterations
Iterations
System Demo
Feature
Feature
Arch
I&A
WSJF
Release Planning
Program Increment
Release on Demand
Develop on Cadence
Feature
Feature
Sprint Goals
Sprint Goals
I&A
WSJF
Release Planning
Program Increment
Architectural Runway
Product Owner
Scrum Master
AGILE TEAMS
Developers & Testers
Code Quality
Agile Architecture
Continuous Integration
Test-First
Stories fit in iterations
Features fit in releases
Spikes, Refactors, Other
TEAM
PROGRAM
v 3.0
Leffingwell, et al.
© 2008–2014 Scaled Agile, Inc.

APPENDIX 3

Interview guide

Interview_Protocol #
Name: Role: Organization:

	General Questions
Q1	Name, Background Education, Position in the organization. Time with the organization? Current assignment with the organization?
A1	Name: Background: Education: Position in the organization: Time with the organization: Current assignment with the organization:
Q2	How many years of experience do you have in software development?
A2	
Q3	How many years of experience do you have in agile software development methods?
A3	
Q4	How long has the organization worked with agile or agile based software development methods?
A4	
Q5	Could you please give a short description of your current project(s), its size and goals? Potential benefits?
A5	

Semi Structured Interview Questions

Q1	How did your organization choose the current software development method (practices) and deploy it? How were the methods implemented? Could you name 5 steps?
A1	
Q2	What are the software development methods based rituals or practices that you encounter in your position? (e.g. Stand-up meetings)
A2	
Q3	Does the organization have any self-created rituals or practices in place? If so, which ones? (e.g. Digital progress monitoring)
A3	
Q4	Is the organization regulated in some way (governance), which prohibits the development teams from using the software development method to full benefit? (allocation of resources)
A4	
Q4.1	What is most regulated?
A4.1	
Q4.2	Do you think that the amount of regulation is appropriate?
A4.2	
Q4.3	What do you think should be regulated and what should be decided by the teams?
A4.3	

Q5	Is the business committed to work together with the developers? What is their reaction on agile and on agile based software development methods?
A5	
Q6	Do you work in close collaboration with business domains of the organization? (e.g. Sales)
A6	
Q7	Do you think that the organization and the software development method are in balance? How, why?
A7	
Q8	Do you think that the organization is using the right software development method? Why? What works well? What could be improved?
A8	
Q9	Do you measure, to ensure that the software development methods is creating benefit for the organization? If so, What is measured? Why?
A9	
Q10	Is the organization (everyone) committed to the software development method?
A10	
Q11	Is the organizational culture appropriate for an agile mindset? Are there some people struggling with agile?
A11	

Q12	What factors prioritize the backlog in the organization? Which parties are involved in this process? Can you generally describe this process?
A12	
Q13	Do you know what your responsibilities are? Deliverables? What are these? As a person and as a team?
A13	
Q14	Do you have any commitment or responsibility outside of your agile team? (E.g. line responsibilities) How do you manage this?
A14	
Q15	Is development initiated by IT or is the business the leading factor for development? Is IT part of business?
A15	

Sentiment analysis Questions

SQ1	What do you know about agile? What level of agile knowledge do you have?										
SA1	1	2	3	4	5	6	7	8	9	10	1 = lowest 10 = highest
SQ2	What is your opinion about agile in general?										
SA2											
SQ3	What do you think, drives the initiative in your organization, regarding agile?										
SA3											
SQ4	Do you think discipline is important?									Yes	No
SQ5	Do you think high level planning is unimportant?									Yes	No
SQ6	Do you think documentation is important?									Yes	No

APPENDIX 4

Interview Table – Shuffled

Structure (Organizational), People and Action

Dedicated teams		People are part of multiple teams and projects	
In dedicated teams, the people are in a fixed team. On the other side, there are people whom are part of multiple teams and projects at the same time.			

Specialist line organization		More cross functional teams	
A cross-functional team is a group of people with different functional expertise working toward a common goal. E.g. finance, marketing and operations. This can include people from anywhere in the organization. Where a specialist line is a group or department with specialized people with the same skill set.			

More flexible governing rules		Management dictates rules	
Teams can adapt the way they are working, by simple governing rules, set by the organization, while in traditional governance the way of working is directed from top down.			

Centralized control		Decentralized decision-making	
Decisions are made in meeting with the right people, with subject experts and on the spot. Degree of decisions autonomy for teams is higher. Centralized control is a more traditional governing structure where the managers give top-down decisions.			

Informal communication		Formal communication	
In traditional structures, there is a degree of formality, and people are only to address their direct superiors. Contact goes to through official channels. Where in agile based organizations, communication is much more informal.			

Shared workspace		Separated workspace	
The development team has a dedicated place where they operate, where separated workspaces, everyone is just working where they are or where they like and are in contact with phone or instant messaging.			

Developers located anywhere		Co-located developers	
Co-located developers, means that all developers are on site, not necessarily sitting together, but are within reach. Developers located anywhere, meaning that your team could be scattered around the world and is communication electronically.			

Small team size		Large team size	
A small team is considered from 5 – 9 people with different roles. Whereas an large team exceeds this number.			

Extended supporting staff		Basic administration as supporting	
The supporting staff in an organization supplies besides the basic administration also additional services and coaching to the operational core. The supporting staff houses a center of experts whom support operations.			

Focus on maintenances (legacy systems)		Focus on updating systems	
The organization focusses on keeping systems up to date by new development. Whereas focus on maintenance results in a situation where the existing software is just maintained.			

Development Process – Requirements, project management, use of methods, participation, training

Iterative process		Linear process	
In an iterative process the team undergoes the same cycle of the project every sprint (2-3 weeks) until the work is complete. Whereas in a linear process, the work is created in one process and benefits and delivering take place at the end of the project.			

Plan-driven - Predictable approach		Value driven - Adaptive approach	
A value driven approach means making decisions or altering decisions if it benefits the value of the product and the process. Plan driven is a more predictable approach where the plan is executed.			

Incremental delivery (continuous)		All in once delivery	
Incremental delivery means that the delivered software grows with each increment. Users can get to know the software feature for feature whereas with all in once delivery, the entire program is delivered all in once.			

Future Value		Direct Value	
Direct value sacrifices documentation and risk analysis, to supply direct value by faster creating working products. Future value has more documentation and risk analysis, but the value for the business, is created in a later stage.			

Test after project completion		Test every Iteration	
Testing every iteration results in issues getting noticed much earlier, when the knowledge of the development is still present in the teams. This results in slower development. Testing when all the code is done can result in problems, from which the origin is unknown, and the bug needs to be find in far more code.			

Interactive input from customer - High client involvement		Predefined user requirements - Low client involvement	
During development iterations, the customers sees the software being created, the customer can speak-up for additional input, so the software is created according to up-to-date input from the customer. With predefined user requirements, the user sees the software when it's done, the result will be, what was specified at the beginning of the project.			

Low refactoring cost		High refactoring cost	
Because the customer is involved at each iteration of the project, changes can be accommodated in an much earlier stage. This has low cost involvement. However, if the product is finished and changes need to be made, the supplier needs to start their development team up again to change something, which could be embedded in the program already.			

Deliver Fixed Scope		Allow change in scope	
Because the customer is involved in the project, the scope of the project can change, due to the fact that the customers realizes that the software in development, will only fix some of their issues, and not all. This choice can translate into a longer project, but with more value to the customer.			

Management provides estimates		Teams are integral part of estimation	
When keeping the team as part of the estimations of progress and planning, a more accurate estimation can be produced, due to the fact that the team is closer to the actual product. Management however, have more insight in future planning and know how long the estimates should be to meet deadlines.			

Continuous improvement each iteration cycle		Lessons learned improvement after project	
After each iteration, retrospectives can be used to improve the process during the project. Retrospectives can also take place at the end of a project, to improve the next project.			

Formal education only		Learning through coaching, workshops, games	
Through coaching, workshop and games, the organization can learn the teams new methods in an interactive and fun manor. Formal education however, teaches new methods in an more formal setting.			

Project Content – Project characteristics, project scope, goals & objectives, resources, technology

Small project size		Large project size	
Small project size means that projects are defined into smaller projects, to make them manageable and better plan able. When working with small teams, delivering small pieces of a smaller project, which in itself can bring value to the organization.			

Low amount of documentation		High amount of documentation	
Due to low documentation, the developers need to meet up to further define their work. With each other or with customers. With more documentation, all requirements are clear, but creating the documents takes more time.			

Off-the shelf software		Customization and in-house development	
Customization and in-house development is a way to personalize software within the organization. This can result in creating additional benefit to the software, this requires developing next to managing the software. Off-the shelf software does not require development and just management. However, the off-the shelf software may not supply all functions required from the software.			

Culture - (Institutional Context) Organizational properties and environmental conditions

Leadership and collaboration		Command and control	
The project team is being supported by certain roles in the organization, e.g. Project manager removes any barriers hindering the core Agile teams. Developers have suggestion they can put forward, due to that they are the experts. With command and control it is execute the work that is specified.			

Competitive environment		Collaborative environment	
Knowledge is shared due to collaboration to innovate work. Within teams the knowledge is shared and collaborative ways of working are used to gain the best results from anyone in the team. In an competitive environment, people can be triggered to perform at their best. Trying to think of new ways to improve and outsmart competitors.			

Management commitment and collaboration		Management as controller, not involved in content	
Management commitment and collaboration results in the involvement of management with the development teams. Whereas management as an controller, the management control the development team from an managerial position, management is not involved in with the development team and the gap in content between the teams and management is larger.			

Management is dictating ways of working		Management is encouraging improvements	
Management is encouraging improvements made by the development team. This means that the development teams has the autonomy to improve their ways of working by themselves. Whereas management is dictating the ways of working, the teams does not have the autonomy to decide their own ways of working and governed by the methods set by the management.			

Transparency of goals, resources and progress		Unclear goals, resources and progress	
Transparency of goals, resources and progress means that everyone related to the project in some manner should be able to know or at least access these items. If this is not the case, if a business member cannot access this data, for the business, it's unclear. Unclear goals, resources and progress can also be related to management keeping this data unclear as a means to control information.			

Strategy and vision is embraced by the entire organization		Strategy and vision is something for top-managers	
Strategy and vision is something that is decided at top-management level and is used by them to guide the organization accordingly. Managers and operations know little about the strategy and vision of the organization and don't have to. Strategy and vision are considered to be something that that top-management should be bothered with.			

Decisions are made based on metrics		Decisions are based on opinions	
Decisions are made based on metrics, meaning that the management is using metrics to make decisions. Whereas decisions are based on opinions.			

Boards and comities act as teams		Boards and comities are political playgrounds	
Boards and comities act as teams, they work together to strive to create the largest value for the organization. Boards and comities can also become political playgrounds where politics dictate the preferences and the outcome.			

ABOUT THE RESEARCHER

To gain a better perception of the paper, we introduce the researcher. If the reader knows the researcher and his background, some chooses and statements may become clearer.

Coen Vermeij is a student from the University of Leiden and the author and researcher of this paper. He is enrolled in the ICT in Business program from the Leiden Institute for advanced Computer Science. The researcher has a background of IT and business studies, over different educational levels. The researcher's knowledge is founded on the basis of an IT study on General IT management and application development. This studies is continued by a bachelor degree in Business, IT and Management from the Avans University of applied sciences, with a minor in Innovation and Technology.

When the researcher finished his bachelor degree, he wanted a new challenge in the form of a master degree. The next step would be the Master program of ICT in Business, due to the similarities in the programs from the bachelor and master. The program ICT in business continues where the bachelor program ended.

Due to the diversity in the different levels of education, the researcher looks at IT from multiple angles. The researcher has development skills, which means that he can communicate with developers in their own terminology. Combining this with the theoretical knowledge from the business aspects of the programs, the researcher is able to talk to both the business and the more technical people related to IT.