Leiden University
Master ICT in Business

Applicability of Agile Scrum within IT infrastructure projects

Name: Stanley Alberts

Date: June 2015

1st supervisor: Dr. H.T. Le Fever

2nd supervisor: Dr. C. J. Stettina

MASTER'S THESIS

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

MASTER'S THESIS

Applicability of Agile Scrum within IT infrastructure projects

Stanley Alberts
Stanley.alberts@kpnmail.nl
S1341944

In partial fulfilment of the requirements for the degree of
Master of Science (M.Sc.) of ICT in Business

Gradution: June 2015

1st supervisor: Dr. H.T. Le Fever
2nd supervisor: Dr. C.J. Stettina

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

Rabobank Netherlands
Portfolio Continuity & Infrastructure
Laan van Eikenstein 9
3705 AR Zeist
The Netherlands

Foreword and Acknowledgements

The document before you is written in completion to the Master's programme ICT in Business followed at Leiden University. This thesis is the result of my research into the applicability of project management Methodology Agile Scrum within IT infrastructure projects. It is written within the context of **Rabobank Netherlands**.

Writing this thesis has been a very instructive experience. The valuable knowledge and practical insights given by several colleagues will be useful in my further career.

For this reason I would like to express my appreciation and acknowledgement to the people who have contributed to the completion of this master thesis.

In the first place I would like to thank my manager, Maud Dull, for giving me the opportunity to carry out this research. Furthermore I would like to thank all colleagues who have provided me with valuable input for this research whereof three colleagues in particular, Anne Giessen André van Heusden and Peter Frederiks, for their guidance and feedback.

In addition, I would like to thank my first supervisor Dr. Hans Le Fever, for his recommendations and guidance during the whole process of writing this thesis.

Last but not least, I would like to thank my parents, brother, family and friends for their continued support during my academic career.

June, 2015

Stanley Alberts

MANAGEMENT SUMMARY

Motivation

Coping with changing customer requirements and achieving a faster time-to-market are of crucial importance in today's fast changing business environment. This has resulted in a shift within the software industry from traditional **process-oriented** methods to more **product-oriented** Agile methods. However, within IT infrastructure projects this shift is not so evident. This raises the question: *To which extent can Agile Methodology Scrum be used as delivery framework within IT infrastructure projects?*

Goal and research method

This research will focus on discovering the applicability of Agile Methodology Scrum within IT infrastructure projects. Through a literature study four characteristics are identified that are typical for Agile Scrum projects. Subsequently, four types of IT infrastructure projects of the **Basic Infrastructure** and **Workplace** programmes are validated on those characteristics through a survey. Those four types of IT infrastructure projects include: Life Cycle Management, Creating New Infrastructure Services, Improving Existing Infrastructure Services and Research Projects. Finally, the data has been analysed via SPSS.

Theoretical framework

Scrum is a popular iterative software development method. The final product grows after every iteration, a so-called sprint. It has been found that four characteristics are typical for Agile Scrum suitable projects.

1. **Divisibility**: By continuously delivering **functional product increments** of the final product, within a sprint up to 4 weeks, a faster time-to-market and continuous feedback of the end-users will be achieved *(Larman, 2004: 9 – 10)*.
2. **Unclear customer needs**: **Changing requirements**, caused by an unclear customer needs, can be managed easily due to the iterative nature of Scrum *(Rubin, 2013; Irshad Kahn et al, 2011; Schwaber, 1997)*.
3. **Stakeholder independency:** Independency of internal and external parties, that pursue their **own goals**, provides a close collaboration within Scrum teams *(Chin, 2004: 17 – 18)*.
4. **Small team size*:*** Small self-organised **dedicated** multidisciplinary teams up to **10 people** provide a close collaboration and effective communication within Scrum projects *(Boehm & Turner, 2004: 28,55; West, D.)*.

Results

Through a survey **62 IT infrastructure projects** are examined. The results are only partially the same as the four characteristics that have been defined in the theoretical framework. Only **3.2%** of the projects met all the four characteristics.

1. **Divisibility**: In only **24,2%** of the projects examined, functional product increments of the final product could be delivered within a sprint up to 4 weeks.
2. **Unclear customer needs**: In only **16,1%** of the projects examined, customer requirements where not clear from the beginning of the project.
3. **Stakeholder independency:** In only **11,3%** of the projects examined independency of internal- and external parties exists that pursue their own goals.
4. **Small team size*:*** In **85,4%** of the projects examined, the team consists of up to 10 people.

Conclusion

With **37,1%,** IT infrastructure projects are dominated by Life Cycle Management projects that are aimed at keeping **existing** infrastructure services in support. This research demonstrated a significant difference between the characteristic **unclear customer needs** and Life Cycle Management projects. In only **4,3%** of the Life Cycle Management projects customer needs were unclear. Usually the customer need for those projects is nothing more than preserving existing functionality. Furthermore the importance of continuity requires a direct delivery of the final product. This can be explained by the significant difference found between the characteristic **divisibility** and the four types of IT infrastructure projects. In only **4.3%** of the projects functional increments of the final product could be delivered. Agile Scrum does not seem an appropriate method here.

Agile Scrum is better suited for IT infrastructure projects that have a closer involvement with application development. These projects are aimed at creating new- and improving exiting IT infrastructure services. Research demonstrated a significant difference between the characteristic **unclear customer needs** and Creating New Infrastructure Services projects. This means that for IT infrastructure project teams customer needs within those projects are more often unclear. In **31,6%** of the Creating New- and **21,4%** of the Improving Existing Infrastructure Services projects customer needs were unclear. In addition a significant cohesion between the characteristic **unclear customer needs** and **divisibility** is demonstrated. Meaning that functional increments of the final product can be delivered more often for IT infrastructure projects where the customer needs are unclear. The data shows that in **42,1%** of the Creating New- and **42,9%** of the Improving Existing Infrastructure Services projects functional increments of the final product could be delivered.

But there is a side note here. A close collaboration between IT infrastructure teams and required specialists has not been established. Continuous availability of specialists is a prerequisite to deliver functional increments of the final product successfully in sprints.

These conclusions are based on statements given in the focus group, conversations with multiple managers and open comment fields of the survey. Research into these statements has not been carried out. However, they clearly support and explain these results.

TABLE OF CONTENT

1 INTRODUCTION

1.1 Problem description

A faster time-to-market, high level of flexibility, high quality and lower costs are of crucial importance in today's fast changing business environment *(Han & Ma, 2015)*. These factors have an influence on project management methods that are used today. They have resulted in a shift within the software industry from traditional **process-oriented** methods to more **product-oriented** Agile methods *(Malliga, 2011)*. Initially, Agile methods were only used for software development projects. Currently more and more organisations see an opportunity to benefit from Agile methods such as Scrum, by applying these methods in other disciplines *(Rubin, 2013)*. The IT infrastructure department of a **Rabobank Netherlands** currently uses the traditional project management method Prince 2 for all their projects. The IT infrastructure department is part of Rabobank's IT organisation that consists of 3.200 FTE. Management would like to respond more effectively on today's fast changing business environment by making use from the advantages of Agile methods. However, within IT infrastructure projects this shift is not so evident.

1.2 Research questions

The main research question is:

> ***" To which extent can Agile Methodology Scrum be used as delivery framework within IT infrastructure projects?"***

In order to answer the main research question the following sub-questions are formulated:

- ***Which project characteristics are typical for an Agile Scrum project approach?***
- ***Which IT infrastructure projects are suitable for an Agile Scrum approach?***

1.3 Research relevance

From an **organisational perspective** this research gives insight into the applicability of project management methodology Agile Scrum within IT infrastructure projects. In literature a lot has been written about the use of Scrum within Software Development projects. However, there is very little literature available about the applicability of Agile Scrum within IT infrastructure projects. From a **scientific perspective** this research provides a small contribution in the field of Agile Scrum and IT infrastructure projects.

1.4 Research scope

As indicated earlier, this research focuses on the applicability of Agile Scrum within IT Infrastructure projects. Scrum is by far the most used Agile method and a lot of literature is available on this subject. *(VersionOne, 2014: 4)*. Based on these facts the choice was made to investigate the applicability of Agile Scrum within IT infrastructure projects.

The following scope has been identified in this research:

- The research takes place within the programmes **Basic Infrastructure** and **Workplace**. Only projects within these programmes will be examined because together they are responsible for all IT Infrastructure projects within the organisation.
- This research focuses on the applicability of Agile Scrum on **project level**. The conditions that must be met to successfully implement Agile Scrum is out of the scope.
- This research is conducted within the context of **Rabobank Netherlands**. IT infrastructure environments within other organisations are out of the scope.

2 PROJECT MANAGEMENT

2.1 Agile versus traditional

Within traditional project management methods such as Waterfall, the final product will be developed by a sequential progression of phases. It is practically impossible to go back to a previous stage *(Moussault et al, 2011: 29)*. After determining the product requirements the customer is less involved in the development process. These methods are very suitable for projects where customer needs and thus the final product can be defined completely *(Chin, 2004: 13 – 15)*. Traditional methods are mainly focused on documentation, a strict planning and detailed design *(Irshad Kahn et. Al, 2011: 442 - 445)*. Schwaber states that the biggest issue of (heavyweight) traditional development methods is the sequential progression of phases *(Schwaber, 1997)*. Dealing with changing customer needs is for this reason difficult. The Waterfall process is shown in figure I.

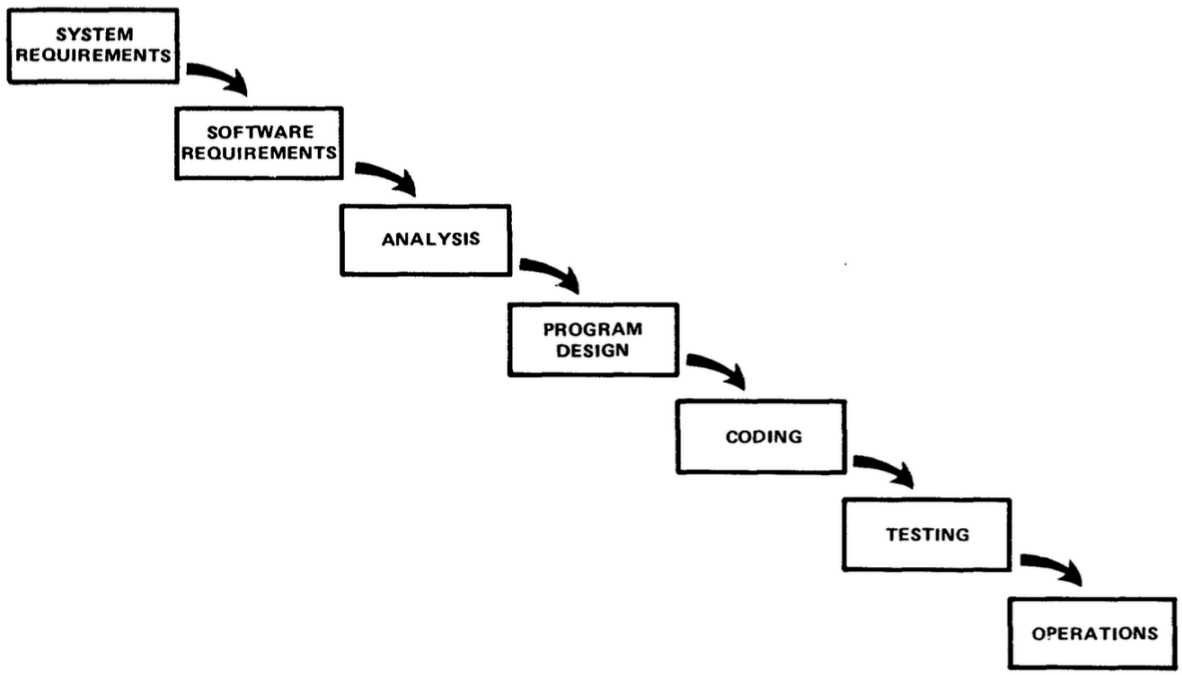


Figure I: *Waterfall model (Royce, 1970)*

Business environments and hence IT projects have changed and become more complex. **Agile project management methods** such as Scrum and Extreme Programming arose as a response to changes in customer behaviour, competitor initiatives, technology breakthroughs, internal organisational changes and the high degree of complexity within software projects *(Ovesen, 2012: 2)*. These (lightweight) development methods are focused on iterative / incremental development of the final product *(Irshad Kahn et. Al, 2011: 442 - 445)*. Lightweight Agile methods are introduced as flexible methods that can deal with changing business requirements. Organisations use these methods in order to respond effectively to changing customer needs *(Chin, 2004: 92 – 94)*. As Dr. Malliga describes in her paper, Agile Software Development have resulted in a shift within the software development community. Where traditional **process-oriented** methods dominated the software industry for many years, the

lightweight **product-oriented** Agile methods became more popular *(Malliga, 2011: 399)*. Agile methods are extremely suitable for projects where the product that needs to be developed is subject to changing business requirements and the final product of these projects cannot concretely be defined *(Chin, 2004: 92 – 94)*.

Agile can be seen as an umbrella term that represents a number of software development methods that uses iterative and incremental development techniques *(Unhelkar, 2013: 32)*. Craig Larman describes iterative development as an approach to develop a product whereby the cycle consists of sequential iterations. An iteration can be seen as a mini-project consisting of activities such as design, implementation, integration and testing. The final product grows after every iteration because functional product increments are added continuously. Continuously adding these functional product increments is called incremental development *(Larman, 2004: 9-10)*.

Compared to traditional methods, the planning process within Scrum is very short because requirements are subject to change *(Mike Beedle et al, 1999: 16)*. The context and the final product that will be delivered is fixed within traditional methods. Within Agile methods the final product is broadly defined, and it is developed based on environmental factors *(Schwaber, 1997: 9)*.

Iterative development techniques were already applied in the 70's and 80's but due to low capacity it was not further developed. The rise of Agile methods started in the mid 90's, when software development methods and technique such as Scrum and Extreme Programming evolved *(Larman & Basili, 2003: 47-56)*. All Agile methods are based on 4 values and 12 principles included in the **Agile Manifesto** which was introduced in 2001, see appendix I *(Highsmith, 2001)*.

Compared to traditional methods, Agile methods are more focused on implicit knowledge, and its hard to transfer this knowledge to other persons. Agile methods gain much agility by sharing this implicit knowledge within the team *(Boehm & Turner, 2005; Fowler & Highsmith, 2001)*. Transfering this implicit knowledge is very expensive because it must be done on the job through close collaboration, coaching and direct communication *(Lam, 2000)*. It is difficult to learn Agile methods from books because within Agile project management methods much is based on experience *(Eysenck & Keane, 2000)*. The Agile transformation process is a very complex organisational transformation. It is much more than just redirecting existing tools and processes *(Nerur, Mahapatra, & Mangalaraj, 2005)*.

2.2 Agile Methodology Scrum

The Agile Framework Scrum as developed by Ken Schwaber and Jeff Sutherland is with 52 percent the most popular Agile method used within western organisations *(VersionOne, 2014: 4)*. Based on this fact in combination with the considerable amount of available literature makes Scrum the perfect Agile method for this research. As previously described, Scrum is an iterative approach regarding software development that is based on Agile principles included in the Agile Manifesto *(Blankenship et al, 2011)*. In addition, Scrum is described as a light development method that provides complete insight, quick adaptability and working within small, dedicated autonomous and self-organised teams *(Leffingwell, 2001)*. Scrum is based on the fact that a system development process is unpredictable and complicated where the final product can only be defined in general *(Schwaber, 1997: 1)*. The following values form the basis of all the work performed by using the Scrum Framework and are a foundation for the Scrum process: focus, courage, openness, commitment and repect. *(Scrum Alliance, 2015; Schwaber & Beedle, 2002: 147)*. A description of these values is included in appendix II.

2.3 Scrum process

The Scrum framework consists of roles, activities and artifacts that together form the Scrum process, shown in figure II. Within this process, the Scrum team is formed by the following roles: **Product Owner**, **Scrum Master** and **Development Team**. Other roles may also be defined but only these are included within the framework. The dedicated team performs prioritized tasks in order to develop the product. Development work is carried out in iterative **sprints** lasting between 1 and 4 weeks. At the beginning of a sprint the team decides which functional increments of the final product they want to deliver and the end of the sprint **(Sprint Backlog)**. These components are selected from a list of items **(Product backlog),** which is prioritized by the Product Owner based on stakeholder input.

The prioritisation will change during the project and is determined based on environmental changes *(Schwaber, 1999: 11)*. Adjustments can't be made after a sprint is started. A sprint will finish on a fixed date regardless if all tasks are fulfilled *(Rubin, 2013)*. The idea is that after every sprint a functional increment that is part of the final product, may be delivered or showed to the customer *(Mike Beedle et al, 1999: 12)*. This involves the delivery of visible and functional product increments. Every product increment builds on the previous *(Rising & Janoff, 2000)*.

In order to communicate the status the activities within the sprint a **Daily Scrum** is held every day. An important part of Scrum is "Inspect and Adapt". At the end of each sprint a **Sprint Review** take place. This review focuses on the developed functional product increment. During this review the functional product increment is demonstrated to stakeholders. In addition, a **Sprint Retrospective** takes place after every sprint. This retrospective focuses on inspecting and adopting the development process *(Rubin, 2013)*. A more detailed process description is included in appendix III.

Figure II: *Scrum process (Scrum Alliance, 2015).*

2.4 Benefits of Scrum

The focus of Scrum on quickly delivering, integrated, tested and valuable business functional product increments every iteration (sprint) ensures for a better return on investment, faster time-to-market and higher customer satisfaction. Additionally, this short-cyclical way of working by continuously delivering functional product increments of the final product causes for less waste in the development process, resulting in a cost savings. The Scrum process provides flexibility. Due to this flexibility organisations that operate in a complex world can respond quickly to competitors and changing requirements of customers and other stakeholders. In addition, the close cooperation within the team and with the customer provides for continuous feedback, strengthened relationships, increased team effectiveness and trust among the team members *(Rubin, 2013: 6)*.

This is confirmed and complemented by Chin. He argues that Scrum is introduced as a flexible method that can adapt to changing business requirements. Because of the iterative nature that causes flexibility, the team is able to respond quickly to changes. In addition, this flexibility minimizes the risks that may occur due to these changes. In an uncertain business environment like today, Agile Scrum is the best method to respond to constantly changing requirements *(Chin, 2004: 92-94)*.

2.5 Scrum in non-software environments

Although Agile Scrum was created for use within software development projects, this method is also applied to other types of projects like hardware development, marketing programmes and sales activities *(Rubin, 2013: 3)*. This is confirmed in an interview with several founders of the Agile Manifesto, as Ron Jefferies mentioned:

> *"Agile ideas are based on how people and organizations work best. There are specialized details we need to know regarding software, just as there are in any other domain. The principles, however, are broadly applicable." (Jackson, 2012: 61)*

Also in this interview Jim Highsmith said the following:

> *"In any project that faces uncertainty, complexity, volatility and risk, there is a place for agile practices and principles." (Jackson, 2012: 61)*

Andrew Hunt concluded with:

> *"At its heart, an agile approach has little to do with software; it's all about recognizing and applying feedback. The definition of "agile" that Venkat Subramanium and I proposed in 'Practices of an Agile Developer' states, "Agile development uses feedback to make constant adjustments in a highly collaborative environment". Notice there is nothing in there about software." (Jackson, 2012: 62)*

Beside the theory, published case studies *(Carlson en Turner, 2013: 469-474)* show that Agile methods can be used in non-software environments, including hardware development. As can be derived from the above-mentioned quote from Jim Highsmith there, he indicates that projects that meet certain characteristics are best suited for an Agile methodology such as Scrum.

2.6 Agile Applicability Models

In order to determine whether an Agile method is a suitable project management method several models (based on project characteristics) are described in literature. Due to the academic background of this research only academic models are used.

2.6.1 Scale model –Alistair Cockburn

The Scale model of Alistair Cockburn is used on the basis of two characteristic **team size** and **project criticality**, to determine if a (light) Agile or (heavy) Traditional development method is most suitable. The model is shown in figure III in the appendix.

- With the characteristic **team size** Cockburn states light development methods are more suitable for projects with small teams where less communication tools are necessary. Agile methods are most suitable for dedicated project teams up to 10 persons. The implementation of Agile principles such as tight coordination and sharing knowledge is difficult for teams between 10 and 30 persons. Agile projects with large teams (30 or more persons) need traditional plans and specifications in order to deal with increasing complexity and multi-dimensional interactions between the project elements *(Boehm & Turner, 2004: 28,55)*.

  This is confirmed and supported by Dave West. He argues that an Agile teams consist of all persons who are needed to deliver functional product increments of the final product. Persons that are working in multiple Agile project teams provide context switching which

poses a risk on the focus of the project. Having one shared goal is important within Agile teams *(West, 2011)*.

- The characteristic **project criticality** is aimed at the impact of an error within an application. Because life critical applications are often related to strict procedures and legal restriction (that require lots of documentations) Cockburn states that Agile methods are not suitable for these applications. The characteristic project criticality is divided in the following classifications: *loss of comfort, loss of discretionary money, loss of irreplaceable money,* and *loss of life*. Within a *loss of comfort* situation Cockburn understands the loss of functionality, which ensures that people can work less comfortable. A *loss of discretionary money* means that money or related value is lost but only in the sense of discomfort. An example of *loss of discretionary money* is an error in a billing system. *Loss of irreplaceable money* refers to money that is lost and cannot be recovered, due to a system error. A *loss of life situation* refers to a system error that actually causes for life-threatening situations. An error within an aircraft control system is an example of such a situation *(Cockburn, 2000)*.

2.6.2 Radar Model – Boehm & Turner

As you can see in figure IV included in the appendix, the Radar model of Boehm & Turner consists of five characteristics that are decisive for an Agile project approach. The **five characteristics** are: **team size**, **criticality**, **dynamism**, **personnel** and **culture.** Boehm en Turner propose that a risk assessment is necessary for projects that score positive on four of the five Agile characteristics. A mix of Agile and traditional methods could be a possibility in this situation.

Of these five characteristics, **team size** and **criticality** are similar to these included in the model of Alistair Cockburn. The characteristics **culture**, **dynamism** and **personnel** are described below.

- The characteristic **culture** reflects the reality that Agile methods are better suited for projects that have to deal with a changing environment. However if the environment is too chaotic and an immediate solution is necessary an Agile approach is not appropriate. In orderly situations strict plans can be followed, where traditional methods are better suited.
- With the characteristic **dynamism** Boehm & Turner state that Agile methods can be best applied in situations where requirements will frequently change. Due to the iterative nature of Scrum changing requirement can be managed easily. However, they don't argue that an Agile approach cannot be used in a situation with little to non-changing requirements. But process-oriented traditional methods aimed at following a plan are better suited for these projects.
- The characteristic **personnel** refers to the expertise of the employees in the project team. Both low- and high expertise levels can be used for projects where a Traditional method is used. Within Agile methods a broader mix of high-level developers is required because teams are

self-organised and functional product increments need to be delivered within a sprint up to 4 weeks *(Boehm & Turner, 2004: 54-57)*.

2.6.3 Chin Model

2.6.3.1 General

Chin describes in his model **two characteristics** that can be applied to select a proper project management method namely: **type of project environments** and **type of organisational stakeholders**. The Chin model is shown as figure V in the appendix.

As Chin mentioned in his book, changing project requirements are derived from uncertainty. And precisely in a world of changing requirements the iterative nature of Agile methods such as Scrum are very useful *(Chin, 2004: 4)*. In addition, Chin states that projects that are carried out in an environment of high uncertainty will benefit the most from Agile methods as compared to projects that are predictable with little to no uncertainty *(Chin, 2004: 13)*. Uncertainty and changing project requirements are inextricably linked. Chin included the facets uncertainty and changing project requirements in the characteristic **type of project environments**.

Chin distinguishes two types of project uncertainty namely: internal- and external uncertainty *(Chin, 2004: 4)*.

- **Internal uncertainty:** involves factors within the project umbrella that can be more or less controlled by the project manager, including scope, schedule and costs.
- **External uncertainty:** involves factors that are out of the project umbrella, such as the changing business environment, competition and changes in business strategy.

2.6.3.2 Type of project environment

In order to determine the degree of project uncertainty Chin described **three type of project environments** specifically: **operational**, - **technology** and **product/process development environments**.

- Projects within an **operational development environment** occur regularly, are similar to each other and are critical for the day-to-day running of the business. These types of projects are quite standard and the organisation knows how to deal with them because they have experienced many similar projects in the past. In addition, these projects are fairly predictable (less changing requirements) and thus there is a low degree of uncertainty. Due to this low degree of uncertainty, these projects are better served by traditional methods, which are more process oriented.
- Within a **technology development environment** projects are aimed at the development of new technology. These are not projects aimed at new product- or application development but

rather the development of breakthrough technology, upon which future products will be built. Projects in this environment have never been carried out before which results in a high degree of uncertainty. The team need to discover multiple pathways and iterations as it progresses toward its end goal. Due to the iterative nature Agile methods can provide great value for these projects.

- In a **product/process development environment** a project can be very complex within large organisations. Projects within this environment must deal with business- and market uncertainties, with on-going changing requirements, wherefore Agile Scrum is very suitable *(Chin, 2004: 14-17)*.

2.6.3.3 Type of organisational stakeholders

To respond quickly to changing requirements in an uncertain environment, a close collaboration within the team and with stakeholders is necessary. Chin's second characteristic **type of organisational stakeholder** is derived from this. *Internal stakeholders* can refer to system owners, top management and programme management. *External stakeholders* can refer to customers, users, partners, and regulatory organisations *(Rubin, 2013: 171)*.

Based on the dependency of internal- and external stakeholders Chin defined the following **three types of stakeholder situations** in order to determine the applicability of Agile methods: **single organisation**, **multiple organisations** and **single company multiple (internal) organisations**.

- Within a **single organisation** the real customer of the project is the business that is sponsoring it. These projects are entirely undertaken within the company and thus there is no dependency of partners or subcontractors. Situations as these are ideal for creating an Agile project environment. Because everyone pursues the same goal a close collaboration and effective communication can be realised much easier than if you have to deal with multiple stakeholder's organisations that pursue their own goals *(Chin, 2004: 17-18)*.
- Within **multiple organisations** it is not impossible to create a successful Agile environment, however it will be more challenging to develop a close collaboration. Usually, Traditional methods are more suited in these situations because expectations of multiple stakeholders such as partners and subcontractors can be managed very well *(Chin, 2004: 18)*.
- Finally Chin defined a situation where a project is executed in a **single company** but the project team has to deal with **multiple (internal) organisations**. This means that a project takes place within a single company but the business units that are involved function as autonomous organisations and pursue their own goals. Depending on the motivation of managers within the organisation it could be easy or difficult to use Agile methods within these situations *(Chin, 2004: 19)*.

2.7 Cynefin Framework

Rubin defined the characteristic **product complexity** based on the Cynefin Framework of Snowden & Boone. By this he is referring to the complexity of the project from a customer desire perspective. In other words, the influence of external environmental factors on the customer needs and thereby project requirements.

The Cynefin Framework *(Snowden & Boone, 2007)* is useful to understand the situation in which a project operates and helps in determining a suitable project method. This framework defines and compares the characteristics of **five** different **domains**: **Complex**, **Complicated**, **Simple**, **Chaotic**, and **Disorder**. It is shown as figure VI in the appendix. Rubin has defined the following five situations using the Cynefin Framework in order to decide if an Agile Scrum approach is useful or not

- Many customer needs of projects within the **complex domain** are unpredictable. Projects with a creative and innovative approach can be categorised in this domain. High levels of interaction and communication between the end-user(s) and the project team are essential. Innovative new product development falls into this category, as do enhancing existing products with innovative new features. Due to the ability to explore, inspect and adapt Agile Scrum is particularly well suited for projects that fall in this domain.
- For projects within the **complicated domain** multiple solutions are possible, but expert diagnosis is required to figure them out. Within this domain customer needs are completely clear and fixed. They are based on good practices. Although Agile Scrum can certainly work with these problems, it might not be the best solution. The use of Agile Scrum adds value in uncertain situations where requirements are likely to change, however this is not expected for these projects.
- Everyone can see the cause and effect relation for projects within the **simple domain**. This is the domain of the best practices, and often the solution is completely clear. When the facts of the solution for this project will be assessed, a proper predefined solution can be determined. Agile Scrum can be used but it may no be the most efficient method for these types of projects. A traditional method with sequential steps may be more suitable for these types of projects because what should be done is entirely clear beforehand.
- Projects within the **chaotic domain** require an immediate action and quick solution. Scrum is not suitable for these projects because prioritizing work in a backlog and determining what work to perform in the next sprint is not useful in these situations. People tend to interpret and act according to their personal preference for action.
- A project falls into the **disorder domain** if it doesn't fit in one of the other domains. This is not a desirable situation because no one completely understands how to manage these kinds of projects. When a project falls into the disorder domain the way out is to break down the

project into constituent parts and assign each to one of the other four domains. Agile Scrum is not used in this domain *(Rubin, 2007: 6-9)*.

2.8 Conclusion literature research

2.8.1 Agile Scrum Characteristics

The goal is to validate IT infrastructure projects based on characteristics that are typical for projects where Agile Scrum is used. Based on existing academic models the following eight characteristics are identified: **(1) team size** *(Cockburn),* **(2) criticality** *(Cockburn),* **(3*)* dynamism** *(Boehm & Turner),* **(4) personnel** *(Boehm & Turner),* **(5) culture** *(Boehm & Turner),* **(6) type of project environment** *(Chin),* **(7) type of organisational stakeholder** *(Chin)*, **(8) project complexity** *(Rubin).*

An essential ninth characteristic that is not included in existing models is related to the possibility to iteratively deliver functional product increments of the final product within a sprint up to 4 weeks *(Larman, 2004: 9 – 10).* Since the delivery of functional increments of the final product within a software project is always possible this characteristic is not included in existing models. Within this research this characteristic is defined as: **divisibility**.

Four of the above characteristics are related to changing requirements. In addition two characteristics proved not to be relevant to determine if Agile Scrum can be used for an IT infrastructure project. For those reasons a choice is made between the characteristics. An explanation regarding the **merging** and **rejection** of the **characteristics** is given in the following paragraphs.

2.8.2 Combining characteristics

The characteristic **dynamism** described by Boehm & Turner scales changing requirements on the base of monthly-expected percentage. Determining this prior to a project is difficult and quite subjective. The three characteristics **culture** *(Boehm & Turner),* **project complexity** *(Rubin)* and **type of project environments** *(Chin)* describe in fact one type of environment regarding changing requirements. However they all have their own interpretation. As Chin describes there is a positive correlation between changing requirements, uncertainty and thus the environment. In other words, changing requirements are already included in the characteristics culture, product complexity and type of project environment and make the characteristic **dynamism** superfluous.

Boehm & Turner basically apply the same scaling to the characteristic **culture** as Rubin however they are based on percentages. Chin outlines three types of project environments in which a project can be categorised. The content of this characteristic is described on a high level and is limited to three situations. Compared to Chin and Boehm & Turner, Rubin's theory takes into account that there is a possibility a project cannot be divided into only one so called domain. Given the more concrete and objective interpretation I have **chosen** for the characteristic **product complexity** of Rubin.

2.8.3 Rejection characteristics

- The characteristic **personnel** described by Boehm & Turner is related to the quality of the developers in the project team. This characteristic provides content to the earlier mentioned characteristic **team size**. The composition and thus the quality of the project team is determined at an organisational level. For this reason it is not useful to determine if a project is suitable for an Agile Scrum approach on a general level through the characteristic personnel.
- The characteristic **criticality** refers to the impact of a programming error in a system. Agile Scrum is not recommended in situations where a programming error leads to life-threatening situations. After consulting with various managers the conclusion was made that impact of a fault in an IT infrastructure project will not lead to life-threatening situations. In a specific project aimed at the core of the network a fault could cause bankruptcy of the Bank. This can be seen as a similar situation. Because this only applies to a very specific situation and this project does not meet the other characteristics, I have chosen to reject the characteristic **criticality**.

2.9 InfraScrum Suitability Model

Based on the above arguments I decided to validate IT infrastructure projects on the following **four characteristics**: **divisibility**, **product complexity**, **type of organisational stakeholders** and **team size**. In order to emphasize the core of each characteristic the following terms will be used is this thesis:

- **Characteristic 1**: **Divisibility** remains unchanged.
- **Characteristic 2**: Product complexity will be **unclear customer needs**
- **Characteristic 3**: Type of organisational stakeholders will be **stakeholder independency**
- **Characteristic 4**: Team size will be **small team size**

The model shown in figure VII states that Agile Scrum is a suitable method for projects that score on the four values in the centre of the model (indicated by the green line).

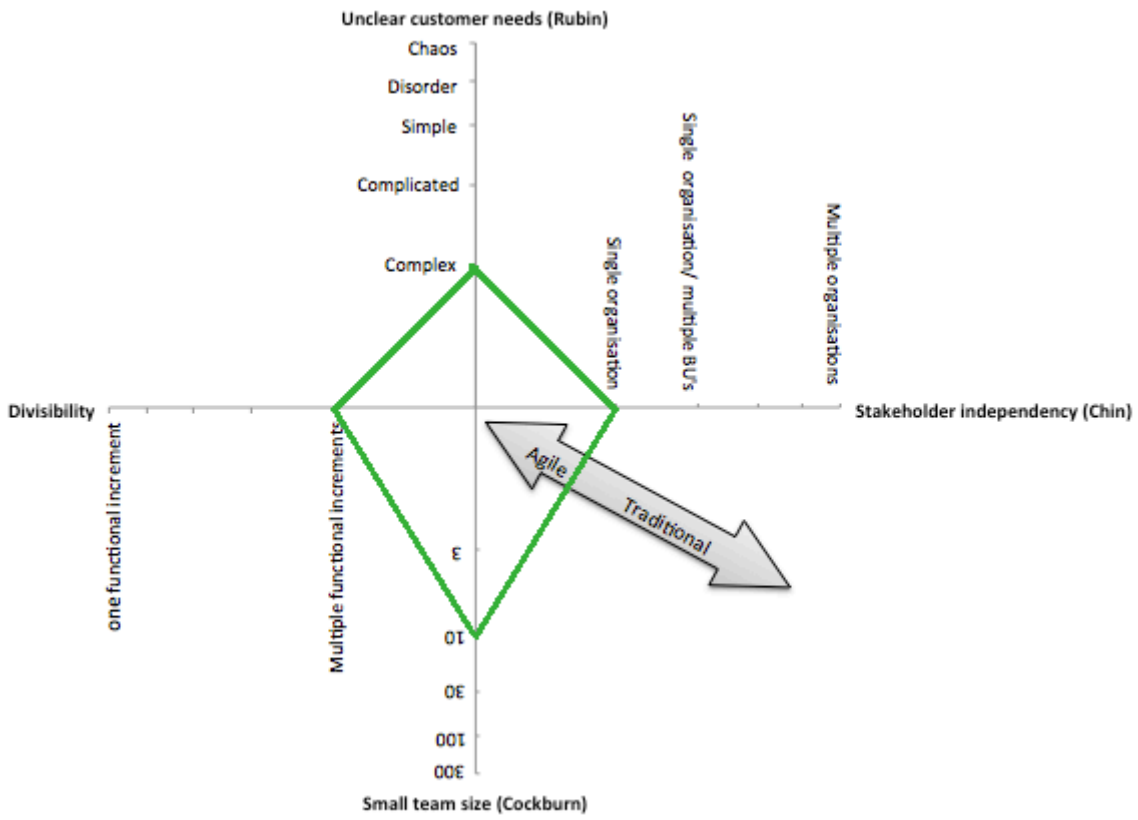


Figure VII: *InfraScrum Suitability Model*

3 IT INFRASTRUCTURE

3.1 Definition IT infrastructure

Despite the fact that IT infrastructures exist for many years a general accepted definition of IT infrastructure doesn't exist. In his book Sjaak Laan describes multiple definitions regarding IT infrastructure. The definitions below provide insight in how IT infrastructure is defined in the literature:

"IT Infrastructure consists of the equipment, systems, software, and services used in common across an organization, regardless of mission/program/project. IT Infrastructure also serves as the foundation upon which mission/program/project-specific systems and capabilities are built" (cio.gov)

"Information technology infrastructure underpins the distributed operational and administrative computing environment. Hidden from the application-based world of end users, technology infrastructure encompasses the unseen realm of protocols, networks, and middleware that bind the computing enterprise together and facilitate efficient data flow. Yet information technology infrastructure involves more than just the mechanics of data systems; it also includes people providing support and services" (Technology Governance Board)

Laan states that what IT infrastructure comprises depends on whom you ask, and what their point of view is. He defined the following **three IT infrastructure perspectives**: **(1) system Manager**, **(2) application user** and **(3) business analyst**. Within this research IT infrastructure will be defined from an **application user perspective**. According to this perspective applications are important for the user but how they are implemented or physically running is invisible. From their perspective this can be seen as IT infrastructure *(Laan, 2013: 33)*.

3.2 IT infrastructure components

Just as there isn't a general accepted definition of IT infrastructure, a general accepted model regarding the construction of an IT infrastructure also doesn't exist. On the other hand IT infrastructure components are described in the literature. According to Broadbent en Weill (1998) an IT infrastructure can be seen as the basic foundation for Information Technology which delivers shared services within the organisation. McKay and Brockway were the first who described IT infrastructure on the basis of a **three-layer model**. Many years later this model has been adapted and elaborated on by Peter Weill. This three-layer model is shown in figure VIII.

- Shared technology components like hardware and software whereof an IT infrastructure exists can be found in the **first layer "IT components"**.

- The **second layer** "**Human IT infrastructure"** is aimed at human- and organisational skills that are necessary to effectively connect IT components in order to create functional IT services.
- Within the **third layer** IT components are connected and function as "**shared IT services"** that support IT applications and business processes. Broadband & Weill (1998) define these shared services as the area of IT infrastructure.

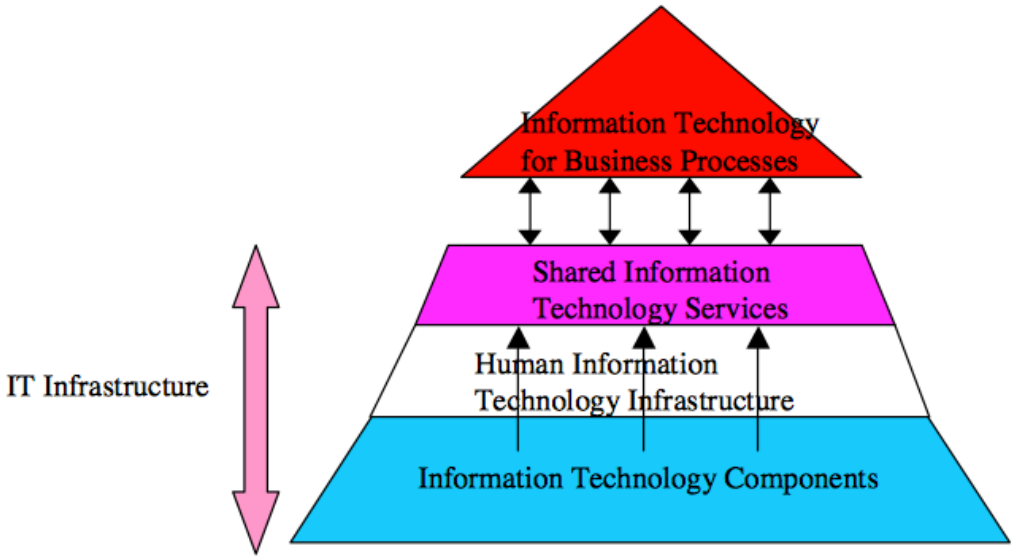


Figure VIII: *Mckay & Brockway – IT-Infrastructure elements*

In fact, an IT infrastructure can be divided into two related components that together provide for a third component shared IT services namely: Human and technical components. Technical components are necessary to share the information capacity of an organisation. The feasibility of the solution depends on the human expertise to set up these components as effectively as possible *(Byrd & Turner, 2000: 169)*.

Laan describes these technical components more concrete and states that an IT infrastructure exists of the following functional building blocks:

- **End user devices:** are the devices used by end users to work with applications, such as PCs, laptops, thin clients, mobile devices, and printers.
- **Operating systems:** are collections of programs that manage a computer's internal workings: its memory, processors, devices, and file system.
- **Storage:** The storage building block contains hard disks, tapes, Direct Attached Storage (DAS), Network Attached Storage (NAS), and Storage Area Networks (SANs).
- **Servers:** are the main physical processing units, computers located in the datacenter.
- **Networking:** is a very important building block of any infrastructure, connecting all components. It includes routers, switches, firewalls, WAN (wide area network), LAN, dial-in, Internet access, and VPNs (Virtual Private Networks).

- **Datacenter floor:** Typically IT hardware is located in a datacenter. Datacenters provide (uninterruptible) power supply, cooling, computer racks, and physical security measures. *(Laan, 2013: 41-42)*.

Despite the fact other authors like, Peter Weill, Jeanne Ross & David Robertson (2006), define middleware as an IT infrastructure component, Laan defined this as a separated layer named "Application Integration Services". In the context of this research middleware and management services will be considered as part of the IT infrastructure.

3.3 IT infrastructure model context research

The model in figure IX has been developed, based on the components as described in the literature. As mentioned before this model is created from an application user perspective. The **green blocks** indicate the field in which the IT infrastructure projects in the context of this research take place.

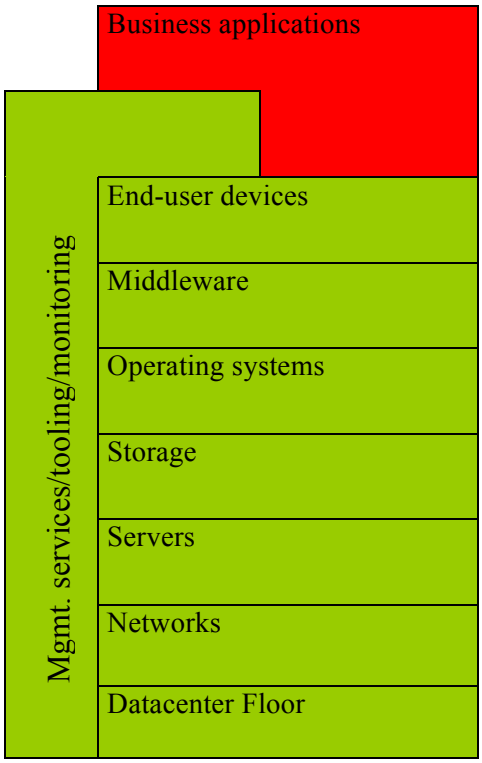


Figure IX: *IT infrastructure Layers*

3.4 IT infrastructure projects

In de context of this research **four types of IT infrastructure projects** are defined:

- **Life Cycle Management projects (LCM):** focus on keeping existing infrastructure services in support. Everything related to new releases/upgrades of software and hardware can be seen as a Life Cycle Management project. An example is the upgrade of an existing operating system.
- **Improving Existing Infrastructure Services projects (IEIS):** focus on the replacement or expansion of existing infrastructure services. Examples of these projects are: the replacement of an infrastructure software tool related to telephone services or the expansion of existing infrastructure-monitoring tools.
- **Creating New Infrastructure Services projects (CNIS):** focus on providing new infrastructure services, both hardware and software. Examples of these projects are the implementation of Microsoft Lync or creating new infrastructure-monitoring tools.
- **Research projects (RP):** focus on the preliminary phase of creating new infrastructure services projects. Based on for example a Proof of Concept or research report you explore if the new feature/product is suitable within the existing infrastructure. Considerations should be made to factors such as security, performance, availability and feasibility.

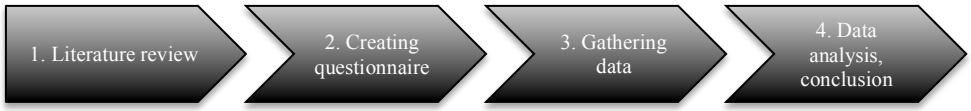
4 METHODOLOGY

4.1 Introduction

This chapter provides a description and argumentation regarding the research strategy of this thesis. Gaining insight into the applicability of project management method Agile Scrum within IT infrastructure projects is the goal of this research.

For this research a quantitative approach is used. The data is gathered via a survey, which provides a practical and structured way to examine a large number of IT infrastructure projects.

An overview of the **four phases** used in this research is shown in the figure below. These phases are elaborated on the following paragraphs.



4.2 Literature study

The literature study focuses on identifying characteristics that are typical for projects where Agile Scrum is used. Databases of *Google Scholar*, *Google Books* and *Leiden University Library* are used to find relevant articles for this research. Combinations of the following keywords are used:

Keywords – *Agile, Scrum; Agile Suitability; IT infrastructure; IT infrastructure model; IT infrastructure components; IT infrastructure projects; Infrascrum.*

4.3 From theory to survey

Through the literature study **four characteristics** that are typical for projects where Agile Scrum is used are identified. Based on the characteristics **divisibility, unclear customer needs, stakeholder independency** and **small team size** a survey was set up. Examination of IT infrastructure projects on these four characteristics is the purpose of this survey.

The survey consists of nine multiple choice questions divided into general- and the four Agile Scrum characteristics. In order to understand the chosen answers regarding question 7 and 9 an open comment field is included. The survey can be found in Appendix IV.

Before the survey was sent the following **three actions** were taken:

- The words **functionality** and **business value** are defined and included in the survey in order to understand concepts behind these words.

- In collaboration with 6 project managers a total of 10 IT infrastructure projects were used to test the survey. A few adjustments were made based on the interpretation of the project managers.
- A presentation for the respondents has been given to explain the background of the survey.

4.4 Data collection

As mentioned in the scope, this research will focus on IT infrastructure projects within the **Basic Infrastructure** and **Workplace** programmes. To make sure respondents were able to answer the questions with actual knowledge, only projects of the **roadmap 2014** were part of this research. This resulted in a list of 79 IT infrastructure projects.

To ensure the statements in this research are based on enough background information IT infrastructure projects are validated via the survey in **two forms**.

1. To examine the projects of the **Basic Infrastructure** programme the survey was sent to 18 project managers.
2. To examine the projects of the **Workplace** programme the survey was used within a focus group session. Five persons participated in this focus group including: one programme manager**,** one architect and three cluster managers. Prior to the session participants were asked for their permission to record the session. The recordings facilitated in making the most concrete possible interpretation of the focus group session.

4.5 Data analysis/conclusion

SPSS Statistics has been used to analyse the data. The following **four** data presentations were made.

- An overview of the scores on the general project characteristics.
- An overview of the scores for each type of project on the Agile Scrum characteristics.
- Correlation analyses were performed to investigate if there were differences between the Agile Scrum characteristics and the four types of IT infrastructure projects.
- Correlation analyses were performed to identify the correlation between the four Agile Scrum characteristics.

The transcription of the recordings of the focus group and text fields of the survey were analysed by using coding. Coding is a derivative of Grounded Theory and used to analyse text *(Corbin & Strauss, 2008)*.

5 RESULTS

An overview of the research results is provided in this chapter. A total of **62 IT infrastructure projects** are examined (response rate 78,5%). General characteristics are described in **paragraph 5.1** followed by a description of the Agile Scrum characteristics in **paragraph 5.2**. After the description of every Agile Scrum characteristic a short conclusion is included. For each type of IT infrastructure project an overview of the scores on the Agile Scrum characteristics is shown in **paragraph 5.3**. This is based on the **InfraScrum Suitability Model** that can be found in **paragraph 2.9**.

The conclusions in this chapter are based on statements given in the focus group, conversations with multiple managers and open comment fields of the survey. An overview of the coding is included in appendix V. To validate those statements multiple SPSS analyses are executed to find significant differences/cohesions. Analyses are performed between the following two types of variables:

1. **Agile Scrum characteristics and the four types of IT infrastructure projects**
   - A *Chi-square test* was performed to find a significant difference between the characteristic **divisibility** and the **four types of IT infrastructure projects**. This test was selected because the characteristic divisibility is the *dependent variable* consisting of nominal/categorical data and the types of IT Infrastructure projects is the *independent variable* with more than 2 groups.
   - A *Kruskal-wallis analysis* followed by a post-hoc analysis from the Scheffe-method was performed to find a significant difference between the characteristic **unclear customer needs** and the **four types of IT infrastructure projects**. This test was selected because the characteristic unclear customer needs is the *dependent variable* consisting of ordinal data and the types of IT Infrastructure projects is the *independent variable* with more than 2 groups.
   - A *Kruskal-wallis analysis* was performed to find a significant difference between the characteristic **stakeholder independency** and the **four types of IT infrastructure projects**. This test was selected because the characteristic stakeholder independency is the *dependent variable* consisting of ordinal data and the types of IT Infrastructure projects is the *independent variable* with more than 2 groups.
   - A *One-way Anova analysis* was performed to find a significant difference between the characteristic **small team size** and the **four types of IT infrastructure projects**. This test was selected because the characteristic small team size is the *dependent variable* consisting of interval data and the types of IT Infrastructure projects is the *independent variable* with more than 2 groups.

2. **Agile Scrum characteristics themselves**
   - A *Spearman correlation analysis* was performed in order to identify the correlation between the **four Agile Scrum characteristics**.

5.1 General project characteristics

The general project characteristics are shown in table I. Of the four examined types of IT infrastructure projects **37,1%** was formed by Life Cycle Management projects followed by Creating New Infrastructure Services **(30,6%)**, Improving Existing Infrastructure Services **(22,6%)** and Research Projects **(9,7%).**

Most of the projects were within budget categories € 100.000 – € 250.000 **(32,3%)** and € 0 – € 50.000 **(27,4%)**. Followed by other categories varying between **6,5%** and **14,5%** of the total.

From the examined projects **19,4%** fell in the project hour categories 1.000 – 1.500 and 1.500 – 2.500 followed by 0 – 500 **(17,7%)**, 500 – 1.000 **(17,7%),** 2.500 – 5.000 **(12,9%)** and 5.000 and more **(12,9%).**

		Frequency	%
Types of projects			
	Life Cycle Management	23	37,1
	Improving existing infrastructure services	14	22,6
	Creating new infrastructure services	19	30,6
	Research projects	6	9,7
Project budget			
	€ 0 - € 50.000	17	27,4
	€ 50.000 - € 100.000	9	14,5
	€ 100.000 - 250.000	20	32,3
	€ 250.000 - € 500.000	8	12,9
	€ 500.000 - 1.000.000	4	6,5
	€ 1.000.000 – and more	4	6,5
Project hours			
	0 - 500 hours	11	17,7
	500 - 1.000 hours	11	17,7
	1.000 - 1.500 hours	12	19,4
	1.500 - 2.500 hours	12	19,4
	2.500 - 5.000 hours	8	12,9
	5.000 – and more	8	12,9

Table I: *general project characteristics*

5.2 Analysis of the Agile Scrum characteristics

The literature study has shown that four characteristics are typical for projects that make use of Agile Scrum. IT infrastructure projects are examined on the basis of the following **four characteristics**: (1) **divisibility,** (2) **unclear customer needs,** (3) **stakeholder independency** and (4) **Small team size.**

The results for every characteristic are described in the following paragraphs. The characteristics are set against the **four types of IT infrastructure projects** that are researched: Life Cycle Management **(LCM),** Improving Existing Infrastructure Services, **(IEIS)**, Creating New Infrastructure Services **(CNIS)** and Research Projects **(RP)**.

5.2.1 Characteristic 1: divisibility

The **first** characteristic **(divisibility)** focuses on the possibility to develop the final product of the project iteratively. By continuously delivering **functional increments** of the final product, within a sprint up to 4 weeks, a faster time-to-market and continuous feedback of the end-users will be achieved *(Larman, 2004: 9 – 10).*

An overview in percentages of the scores on this characteristic per type of IT infrastructure project is shown in table II.

			Divisibility		Chi-Square Tests	
			One functionality	**Multiple functionalities**		
Type of Project	**LCM**	% within type of project	95,7	4,3	Value	12.84
		% within Divisibility	46,8	6,7	df	3
					Asymp. Sig. (2-sided)	0.005
	IEIS	% within type of project	57,1	42,9		
		% within Divisibility	17,0	40,0		
	CNIS	% within type of project	57,9	42,1		
		% within Divisibility	23,4	53,3		
	RP	% within type of project	100,0	0,0		
		% within Divisibility	12,8	0,0		

Table II: *Scores characteristic divisibility versus types of IT infrastructure projects.*

SPSS - Chi-square test

In order to determine whether there are **differences** in scores between the characteristic **divisibility** and the **four types of IT infrastructure projects** a Chi-square test is performed. Because the characteristic **divisibility** is nominal a Chi-square test was selected. The characteristic **divisibility** was used as *dependent* variable and the four types of IT infrastructure projects as *independent*.

Divisibility versus the **four types of IT infrastructure projects***:* there is a **significant difference** between the possibility to deliver functional increments of the final product within a sprint up to 4 weeks and the four types of IT infrastructure projects, $\chi 2\ (3) = 12.84$, $p < 0.05$.

Projects where **functional increments** of the final product **couldn't** be **delivered** within a sprint up to 4 weeks Life Cycle Management projects had the lowest percentage **(46,8%)**, followed by Creating New Infrastructure Services **(23,4%)**, Improving Existing Infrastructure Services **(17,0%)** and Research Projects **(12,8%)**.

Project where **functional increments** of the final product **could** be **delivered** within a sprint period up to 4 weeks Creating New Infrastructure Services had the highest percentage **(53,3%)**, followed by Improving Existing Infrastructure Services **(40,0%)** and Life Cycle Management projects **(6,7%)**. Research Projects have no share in this.

In other words, **functional increments** of the final product can be delivered **more often** for projects in the category Creating New Infrastructure Services than Life Cycle Management projects.

Conclusion characteristic 1: divisibility

In only **24,2%** of the projects examined, functional increments of the final product could be delivered within a sprint up to 4 weeks. Two statements are identified:

- IT infrastructure projects are dominated by projects aimed at keeping **existing** infrastructure services in support (Life Cycle Management). The importance of continuity requires a direct delivery of the final product.
- The throughput time of functional increments of the final product exceeds the maximum sprint duration up to 4 weeks.

The data shows that functional increments of the final product can be delivered more often for projects that have a closer involvement with application development.

5.2.2 Characteristic 2: unclear customer needs

The **second** characteristic **(unclear customer needs**) focuses on anticipating fast changing customer requirements. The iterative character of Scrum is extremely useful for this. For this reason Scrum is used for projects where customer needs are unclear from the beginning of the project *(Rubin, 2013; Irshad Kahn et. al, 2011; Schwaber, 1997).*

An overview in percentages of the scores on this characteristic per type of IT infrastructure project is shown in table III.

			Unclear customer needs (*Rubin)*		
			Simple	**Complicated**	**Complex**
Type of project	**LCM**	% within type of project	82,6	13,1	4,3
		% within unclear customer needs	61,3	14,3	10,0
	IEIS	% within type of project	35,7	42,9	21,4
		% within unclear customer needs	16,1	28,6	30,0
	CNIS	% within type of project	21,1	47,3	31,6
		% within unclear customer needs	12,9	42,8	60,0
	RP	% within type of project	50,0	50,0	0,0
		% within unclear customer needs	9,7	14,3	0,0

Table III: *Scores characteristic unclear customer needs versus types of IT infrastructure projects.*

SPSS – Kruskal-wallis analysis

In order to determine whether there are differences in scores between the characteristic **unclear customer needs** and the types four of IT infrastructure projects a Kruskal-wallis analysis is performed. The characteristic unclear customer needs was used as *dependent* variable and the four types of IT infrastructure projects as *independent*.

Unclear customer needs versus the **four types of IT infrastructure projects:** there is a **significant difference** between the degree of unclear customer needs at the start of the project and the four types of IT infrastructure projects, $\chi 2$ (3) = 16.30, $p < 0.05$. A post-hoc analysis from the Scheffe-method identified that **customer needs** are **more often unclear** at projects in category Creating New

Infrastructure Services (*M* = 2.11, *SD* = 0.76) than by Life Cycle Management projects (*M* = 1.22, *SD* = 0.52).

There was no significant difference between the other projects and the extent of unclear customer needs at the start of the project.

Conclusion characteristic 2: unclear customer needs

In only **16,1 %** of the projects examined, customer needs were not clear at the start of the project. The following statement is identified:

- The creation of the most IT infrastructure projects is related to end of support of the supplier (Life Cycle Management). Usually, the only customer need is preserving existing functionality.

The data shows that customer needs are more often unclear at projects, which have a closer involvement with application development. IT infrastructure projects are dominated by projects aimed at keeping **existing** infrastructure services in support (Life Cycle Management). The importance of continuity requires a direct delivery of the final product.

5.2.3 Characteristic 3: stakeholder independency

The **third** characteristic **(stakeholder independency**) focuses on the importance of a close collaboration and having a **shared goal** within the team. This is important in order to deliver successful functional increments of the final product. For this reason Scrum is used in projects that are independent of internal and/or external parties which pursue their own goals *(Chin, 2004: 17 – 18).*

An overview in percentages of the scores on this characteristic per type of IT infrastructure project is shown in table IV.

			Stakeholder Independency *(Chin)*		
			Single Organisation	**Single organisation / multiple BU's**	**Multiple organisations**
Type of project	**LCM**	% within type of project	8,7	39,1	52,2
		% within stakeholder independency	28,6	34,6	41,4
	IEIS	% within type of project	28,6	50,0	21,4
		% within stakeholder independency	57,1	26,9	10,3
	CNIS	% within type of project	5,3	42,1	52,6
		% within stakeholder independency	14,3	30,8	34,5
	RP	% within type of project	0,0	33,3	66,7
		% within stakeholder independency	0,0	7,7	13,8

Table IV: *Scores characteristic stakeholder independency versus types of IT infrastructure projects.*

SPSS – Kruskal-wallis analysis

In order to determine whether there are differences in scores between the characteristic **stakeholder independency** and the **four types of IT infrastructure projects** a Kruskal-wallis analysis was performed. The characteristic stakeholder independency was used as *dependent* variable and the four types of IT infrastructure projects as *independent*.

Stakeholder independency versus the **four types of IT infrastructure projects:** there is **no significant** difference between the degree of stakeholder independency and the four types of IT infrastructure projects, $\chi 2\ (3) = 5{,}09$, $p = 0.166$.

Conclusion characteristic 3: stakeholder independency

In only **11,3%** of the projects examined were independent of internal- and external parties that pursue their own goals. Two statements are identified:

- In order to create the final product of IT infrastructure project multiple disciplines like network, firewall and storage specialists are momentarily needed. Their main activity is work in the line organization in order to ensure continuity. They perform these tasks next to it.
- In addition, there are some projects were beside internal specialists also external specialists from suppliers are involved.

5.2.4 Characteristic 4: small team size

The **fourth** characteristic **(small team size**) focuses on the size of the team. In order to establish an effective team communication, Agile Scrum project teams are self-organised, dedicated, multidisciplinary and consist of up to **10 people** *(Boehm & Turner, 2004: 28,55; West, D.).*

An overview in percentages of the scores on this characteristic per type of IT infrastructure project is shown in table V.

			Small team size *(Cockburn)*			
			0 – 3	**3 – 10**	**10 – 30**	**30 - 100**
Type of project	**LCM**	% within type of project	13,0	78,3	8,7	0,0
		% within small team size	27,3	42,9	25,0	0,0
	IEIS	% within type of project	14,3	64,3	14,3	7,1
		% within small team size	18,2	21,4	25,0	100,0
	CNIS	% within type of project	15,7	63,2	21,1	0,0
		% within small team size	27,2	28,6	50,0	0,0
	RP	% within type of project	50,0	50,0	0,0	0,0
		% within small team size	27,3	7,1	0,0	0,0

Table V: *Scores characteristic stakeholder independency versus types of IT infrastructure projects.*

SPSS – One-way Anova analysis

In order to determine whether there are differences in scores between the characteristic **small team size** and the **four types of IT infrastructure projects** a One-way Anova analysis was performed. The characteristic **small team size** was used as *dependent* variable and the four types of IT infrastructure projects as *independent*.

Small team size versus the **four types of IT infrastructure projec**t*:* there is **no significant** difference between the size of IT infrastructure teams and the four types of IT infrastructure projects, $F(3) = 1.71$, $p = 0.18$.

Conclusion characteristic 4: small team size

In **85,4%** of the projects examined, the team consists of up to 10 people. IT infrastructure **teams** are **not dedicated** because the dependency of specialists.

5.3 InfraScrum Suitability Model Scores

In these paragraphs an overview of the scores on the Agile Scrum characteristics is shown per type of IT infrastructure project. The **InfraScrum Suitability model** is shown in **paragraph 2.9**. Only **3,2%** of the 62 examined projects met all the **four characteristics**.

SPSS – Spearman analysis

Based on theoretical considerations, a Spearman correlation analysis was performed in order to identify the correlation between the four Agile Scrum characteristics.

Divisibility and **unclear customer needs***:* there is a **significant cohesion** between the possibility to deliver functional increments of the final product within a sprint up to 4 weeks and the degree of unclear customer needs and the four types of IT infrastructure projects, ($r = 0.51$, $p < 0.05$). In other words, functional increments of the final product can be delivered more often when customer needs are unclear. There is no significant cohesion between the other characteristics.

5.3.1 Life Cycle Management projects

An overview of the scores on the four Agile Scrum characteristics of the **23** validated **Life Cycle Management projects** is shown in table VI.

	LIFE CYCLE MANAGEMENT PROJECTS			
	Divisibility	**Unclear customer needs**	**Stakeholder independency**	**Small team size**
Project name	Multiple	Complex	Single organisation	0 – 10
Implementation DB2 11 Banking	✗	✗	✓	✓
NonStop OS Release 2014	✗	✗	✗	✓
LCM UCM PS7	✗	✗	✗	✓
Phase out RHELL 5	✗	✗	✗	✓
Phase out X86 blauwdruk 2.0	✗	✗	✗	✓
LCM-DCN	✗	✗	✗	✓
Connectivity Remote Locations	✗	✗	✗	✓
Scom	✗	✗	✗	✗
SQL upgrade under Sharepoint 2010	✗	✗	✗	✓
SQL 2014	✗	✗	✗	✓
San Upgrade	✗	✗	✗	✓
LCM AppSense	✗	✗	✗	✓
LCM FAX (upgrade + SIP 2012-23)	✗	✗	✗	✓
Video conferencing LCM	✗	✗	✗	✓
LCM endpoint security	✗	✗	✗	✓
Lync 2014	✗	✗	✗	✓
Upgrade AD + ADFS	✗	✗	✗	✓
LCM Application Delivery (streaming)	✗	✗	✗	✓
JAVE Upgrade	✗	✗	✗	✓
IE11	✗	✗	✗	✓
Implementation AppV SP2-HF5	✗	✗	✗	✓
WAS	✗	✗	✗	✗
MQ 7.5 AIX, LINUX & Windows	✓	✗	✓	✓

Table VI: *Scores Agile Scrum characteristics versus Life Cycle Management projects.*

5.3.2 Improving Existing Infrastructure Services projects

An overview of the scores on the four Agile Scrum characteristics of the **14** validated **Improving Existing Infrastructure Services projects** is shown in table VII.

	IMPROVING EXISTING INFRASTRUCTURE SERVICES			
	Divisibility	**Unclear customer needs**	**Stakeholder independency**	**Small team size**
Project name	Multiple	Complex	Single organisation	0 - 10
ECO3: Exadata migration databases	✗	✗	✗	✗
MQ-AMS	✗	✗	✗	✓
Implementatie Oracle standby databases	✗	✗	✗	✓
Autonomie GBI	✗	✗	✓	✓
Centralisation fileservers LB	✗	✗	✗	✓
Deviation fileserver RN	✗	✗	✗	✓
Rabo webinar	✗	✗	✗	✓
Multimedia VDI	✗	✗	✗	✗
SVC expansion	✓	✗	✗	✗
Overview of chains (B=3)	✓	✗	✓	✓
Management Tooling 2014	✓	✓	✓	✓
Expansion Automation	✓	✓	✓	✓
Integral acceptance environment	✓	✗	✗	✓
Phase Get it Done	✓	✓	✗	✓

Table VII: *Scores Agile Scrum characteristics versus Improving Existing Infrastructure Services projects.*

5.3.3 Creating New Infrastructure Services projects

An overview of the scores on the four Agile Scrum characteristics of the **19** validated **Creating New Infrastructure Services projects** is shown in table VIII.

	CREATING NEW INFRASTRUCTURE SERVICES			
	Divisibility	**Unclear customer needs**	**Stakeholder independency**	**Small team size**
Project name	Multiple	Complex	Single organisation	0 - 10
NAS: 2014	✗	✗	✓	✗
New mail environment RI/RN	✗	✗	✗	✓
Greenfield AD	✗	✗	✗	✓
Mobile Integration	✗	✗	✗	✓
BYOD for external staff	✗	✗	✗	✓
Managed werkplace	✗	✓	✗	✓
Implementation Golden Image	✗	✗	✗	✓
RaboCast extern	✗	✗	✗	✗
VDI 3.0	✗	✗	✗	✗
AIX Auditing	✓	✗	✗	✓
Storage automation-taks	✓	✓	✗	✓
Central Log Management	✓	✓	✗	✓
Implementation uCMDB	✓	✓	✗	✓
Implementation SIS app	✓	✓	✗	✓
Ipad in the Bankhal	✓	✗	✗	✓
VDI Monitoring	✓	✗	✗	✓
iMW	✓	✓	✗	✓

Table VIII: *Scores Agile Scrum characteristics versus Creating New Infrastructure Services projects.*

5.3.4 Research Projects

An overview of the scores on the four Agile Scrum characteristics of the **6** validated **Research Projects** is shown in table IX.

	RESEARCH PROJECTS			
	Divisibility	**Unclear customer needs**	**Stakeholder independency**	**Small team size**
Project name	Multiple	Complex	Single organisation	0 - 10
Research EOL MQ Server NSK	✗	✗	✗	✓
Automatic assignment to the correct network	✗	✗	✗	✓
Research storage performance & capacity monitoring	✗	✗	✗	✓
Research KOTA	✗	✗	✗	✓
Research multimedia VDI	✗	✗	✗	✓
Research Cloud	✗	✗	✗	✓

Table IX: *Scores Agile Scrum characteristics versus Research Projects.*

6 CONCLUSION

Based on the results obtained from the survey supplemented with the argumentation of the focus group a concrete answer on the main research question in the context of **Rabobank Netherlands** can be given.

" To which extent can Agile Methodology Scrum be used as delivery framework within IT infrastructure projects?"

With **37,1%,** IT infrastructure projects are dominated by Life Cycle Management projects that are aimed at keeping **existing** infrastructure services in support. This research demonstrated a significant difference between the characteristic **unclear customer needs** and Life Cycle Management projects. In only **4,3%** of the Life Cycle Management projects customer needs were unclear. Usually the customer need for those projects is nothing more than preserving existing functionality. Furthermore the importance of continuity requires a direct delivery of the final product. This can be explained by the significant difference found between the characteristic **divisibility** and the four types of IT infrastructure projects. In only **4.3%** of the projects functional increments of the final product could be delivered. Agile Scrum does not seem an appropriate method here.

Agile Scrum is better suited for IT infrastructure projects that have a closer involvement with application development. These projects are aimed at creating new- and improving exiting IT infrastructure services. Research demonstrated a significant difference between the characteristic **unclear customer needs** and Creating New Infrastructure Services projects. This means that for IT infrastructure project teams customer needs within those projects are more often unclear. In **31,6%** of the Creating New- and **21,4%** of the Improving Existing Infrastructure Services projects customer needs were unclear. In addition a significant cohesion between the characteristic **unclear customer needs** and **divisibility** is demonstrated. Meaning that functional increments of the final product can be delivered more often for IT infrastructure projects where the customer needs are unclear. The data shows that in **42,1%** of the Creating New- and **42,9%** of the Improving Existing Infrastructure Services projects functional increments of the final product could be delivered.

But there is a side note here. A close collaboration between IT infrastructure teams and required specialists has not been established. Continuous availability of specialists is a prerequisite to deliver functional increments of the final product successfully in sprints.

These conclusions are based on statements given in the focus group, conversations with multiple managers and open comment fields of the survey. Research into these statements has not been carried out. However, they clearly support and explain these results.

7 DISCUSSION

In this chapter the research will be evaluated. First of all the limitations of this research are indicated followed by suggestions for further research.

7.1 Limitations and suggestions for further research

- The context in which this research has been conducted is certainly not completely representative for other organisations. Similar research into other organisations would be an interesting study.
- In order to ensure the continuity of existing IT infrastructure services the Rabobank IT organisation has a strong focus on maintaining existing infrastructure services. An in-depth investigation into the causes of this focus is an interesting topic for further research.
- This research has demonstrated that Agile Scrum is better suited for infrastructure projects that have a closer involvement with application development. Since the customer needs for those projects are not clear for IT infrastructure teams an interesting topic for further research would be to investigate how to achieve a closer cooperation between these two parties.

REFERENCES

Unhelkar, B. (2013). *The Art of Agile Practice: A Composite Approach for Projects and Organizations.* Boca Raton: CRC Press.

Larman, C. & Basili, V.R. (2003, June). *Iterative and Incremental development: A Brief History.* IEEE Computer Society. Retrieved from: https://www.cs.umd.edu/~basili/publications /journals/J90.pdf.

Highsmith, J. (2001). *History: The Agile Manifesto.* Retrieved from: http://agilemanifesto.org

VersionOne (2014). *8th Annual State of Agile Survey*: VersionOne Inc. Retrieved from: http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf.

Malliga, T. J. (2011). *Agile Software Development: An approach to light weight from heavy weight.* International Journal of Engineering Science and Technology. Vol. 3, No. 1. Retrieved from http://www.ijest.info/docs/IJEST11-03-01-123.pdf.

Ovesen, N (2012). *The challenges of becoming Agile: Implementing and conducting Scrum in Integrated Product Development.* Retrieved from: http://www.theinventivestep.net/ Ovesen-2012_The-Challenges-of-Becoming-Agile_WEB.pdf.

Irshad Khan, A., Jameel Qurashi, R., & Ali Khan. U. (2001). *A Comprehensive Study of Commonly Practiced Heavy and Light Weight Software Methodologies.* International Journal of Computer Science Issues. Vol. 8, Issue 4, No 2. Retrieved from: http://ijcsi.org/papers/IJCSI-8-4-1-441-450.pdf.

Leffingwell, D. (2007). *Scaling Software Agility: Best practices for large Enterprises.* Boston: Pearson Education Inc.

Blankenship, J., Bussa, M., & Millet, S. (2011). *Pro Agile .NET Development with Scrum.* New York: Apress.

Rubin, K.S. (2013). *Essential Scrum: A practical Guide to the most populair Agile process.* New York: Addison-Wesley.

Jackson, M.B. (2012). *Agile: A Decade in. PM Network. 26 (4).*

Carlson, R. & Turner, R. (2013). *Review of Agile Case Studies for applicability to Aircraft Systems Integration.* Procedia Computer Science.

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

Moussault, A., Baardman E., & Brave F. (2011). *Wegwijzer voor methoden bij projectmanagement.* Zaltbommel: Van Haren Publishing.

Snowden, D.J., Boone, M.E. (2007, November). *A Leader's Framework for Decision Making.* Harvard Business Review.

Schwaber, K. & Beedle, M. (2002). *Agile software development with Scrum.* Upper Saddle River: Prentice Hall.

Rising, L. & Janoff, N.S. (2000). *The Scrum Software Development Process for Small Teams.* IEEE Software July/August.

Scrum Alliance Inc. (2015). *Scrum: Transforming the world of work*. Retrieved from: https://www.scrumalliance.org/scrum/media/ScrumAllianceMedia/Files%20and%20PDFs/About%20Us/2015-Strategy.pdf

Larman, C. (2004). *Agile & iterative development: A managers guide.* Boston: Pearson Education Inc.

Cockburn, A. (2000, July/August). *Selecting a project's methodology.* IEEE Software. Retrieved from: http://www.eee.metu.edu.tr/~bilgen/Cockburn647.pdf.

Boehm, B. & Turner, R. (2004). *Blancing Agility and Discipline: A Guide for the Perplexed.* Boston: Person Education Inc.

Laan, S. (2011). *IT infrastructure Architecture – Infrastructure building blocks and concepts.* Raleigh: Lulu Press Inc.

Kozierok, C.M. (2005). *The TCP/IP Guide: A Comprehensive, illustrated Internet Protocols Reference.* San Francisco: No Starch Press.

Blank, A.G. (2004). *TCP/IP foundations.* Alameda: Sybex Inc.

Laan, S. (2013). *IT infrastructure Architecture – Infrastructure building blocks and concepts.* Raleigh: Lulu Press Inc.

Byrd, T.A., & Turner, D.E. (2000). *Measuring the flexibility of Information Technology Infrastructure: Exploratory Analysis of a Construct. Joural of Management Information Systems.*

Ross, J.W., Weill, P. & Robertson D.C. (2006). *Enterprise Architecture as Strategy: Creating a foundation for business execution.* Boston: Harvard Business School Publishing.

Weill, P. & Broadbent, M. *Leveraging the New Infrastructure: How Market Leaders Capitalize on IT.* Harvard business School Press, May 1998.

Resnick, S., Bjork, A., & Plaza, de la, M. (2011). *Professional Scrum with Team Foundation Server 2010.* Indianapolis: Wiley Publishing Inc.

West, D. (2011). *Water-Scrum-Fall is the reality of Agile for most organizations today.* Cambridge: Forrester Research.

Corbin, J.M. & Strauss, A.L. (2008). *Basics of qualitative research: techniques and procedures for developing grounded theory.* Los Angeles: SAGE Publications.

Lam, A. (2000). *Tacit knowledge, organizational learning and societal institutions: an integrated framework.*

Fowler, M. & Highsmith, J. (2001). *The Agile manifesto. Software Development.*

Boehm, B. & Turner, R. (2005). *Management challenges to implementing agile processes in traditional organizations.*

Eysenck, M. & Keane, M. (2000). *Cognitive psychology: A student's handbook.*

Nerur, S., Mahapatra, R. & Mangalaraj, G. (2005). *Challenges of migrating to agile methodologies.*

Beedle, M., Devos, M., Sharon, Y., Schwaber, K. & Sutherland, J. (1999). *Scrum: An externsion pattern language for hyperproductive software development.*

Schwaber, K. (1997). *SCRUM Development Process.*

Han, J. & Ma, Y. (2015). *Software Project Planning Using Agile.*

.

APPENDIXES

Appendix I: Agile values and principles

Within the Agile mind-set is chosen for:

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

While there is value in the items on the right, we value the bold items on the left more.

The following **12 principles** are included in the Agile Manifesto:

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

Appendix II: Scrum values

The **5 values** below form the basis for the progress of the Scrum Team *(Scrum Alliance, 2015)*:

1. **Focus:** because we focus on only a few things at a time, we work well together and produce excellent work. We deliver valuable items sooner.
2. **Courage:** because we work as a team, we feel supported and have more resources at our disposal. This gives us the courage to undertake greater challenges.
3. **Openness:** as we work together, we express how we are doing, what's in our way, and our concerns are so they can be addressed.
4. **Commitment:** because we have great control over our own destiny, we are more committed to success.
5. **Respect:** as we work together, sharing successes and failures, we come to respect each other and to help each other become worthy of respect.

Appendix III: Scrum Framework process description

The Scrum framework consists of roles, activities and artifacts that together form the Scrum process. Before describing the Scrum process it is important to explain the **3 Scrum roles** that together form the Scrum Team: **Product Owner**, **Scrum Master** and **Development Team**. Other roles may be defined but only these are included within the framework.

The **Product Owner** is responsible for prioritizing the functional increments in the product backlog and communicates a clear vision of the solution to all involved parties. A **Product Backlog** is a prioritised list of the work that needs to be done in order to deliver the final product. The **Scrum Master** helps the team with the understanding, compliance and the applicability of the values, principles and process components within Scrum. Within Scrum the **Development Team** is dedicated, self-organised and determines the best way to achieve the goal that is prepared by the product owner.

The product owner has a vision regarding the final product that needs to be delivered. But the development of the final product is very complex due to changing requirements. For this reason the final product is developed in multiple functional increments that are prioritized in the product backlog.

These functional increments are delivered after a **Sprint** up to 4 weeks. A sprint begins with a **Sprint Planning**, which includes the development work during the sprint. The number of components in the product backlog is more than the team can deliver within a short sprint. Before the beginning of a sprint the team has to select some items from the product backlog that they think they can complete within a sprint up to 4 weeks. This activity is called the sprint planning. To gain confidence that the team has made a realistic choice in selecting the sprint items, a **Sprint Backlog** is made. Within the sprint backlog a number of detailed tasks regarding the fulfilment of the selected product backlog items is described. Subsequently, the sprint execution takes place. In this phase the team performs the tasks necessary to deliver the selected items at the end of the sprint. The performance of the sprint is taken on a daily basis during the **Daily Scrum.** This ensures that everyone is informed of the workflow and problems can be solved quickly. At the end of every sprint the team delivers potential useful product increment(s). Scrum teams complete the sprint by conducting an inspect- and adapt activity. Within the **Sprint Review,** the stakeholders and Scrum team inspect the delivered functional increment. Thereafter the Scrum team evaluates the development process during the **Sprint Retrospective**. The outcome of these evaluation sessions could cause for adjustments in the prioritizing of the product backlog and/or changes in the development process. After these evaluations the whole Scrum process starts again and repeats until the final product is delivered *(Rubin, 2013)*.

Appendix IV: InfraScrum survey

De ingevulde antwoorden per project leiden met behulp van het bijbehorende InfraScrum Suitability model tot een advies met betrekking tot de uitvoerbaarheid van een IT-Infrastructuur project middels Agile-Scrum.

Survey project research

In het kader van mijn onderzoek naar de toepasbaarheid van project management methode Agile-scrum binnen IT-Infrastructuur projecten heb ik deze enquête opgesteld. Het doel van deze enquête is om data te verzamelen omtrent de kenmerken van alle projecten opgenomen in de BIS Roadmap 2014. Op basis van deze data zal ik een analyse uitvoeren wat zal leiden tot een advies aangaande de toepasbaarheid van Agile-Scrum binnen IT-Infrastructuur projecten. Als projectmanager zou ik je willen vragen deze enquête in te vullen voor ieder project waar je in 2014 verantwoordelijk voor was.

Het onderzoek bestaat uit 9 meerkeuzevragen welke opgedeeld zijn in twee categorieën te weten: algemene project kenmerken en kenmerken gerelateerd aan de toepasbaarheid van Agile-Scrum. Het is niet mogelijk meerdere antwoorden aan te kruisen. Kies het antwoord wat het meest toepasselijk is voor dit project.

Mocht je nog vragen of opmerkingen hebben tijdens het invullen van deze enquête dan ben ik bereikbaar in Zeist, locatie oranje 1, telefonisch 06-46737228 of per e-mail op s.alberts@rn.rabobank.nl.

Bij voorbaat dank voor je medewerking aan dit onderzoek!

Om de bij vraag 9 gebruikte begrippen ***functionaliteit*** en ***business value*** te verduidelijken zijn deze hieronder gedefinieerd.

Toelichting opgenomen begrippen:

- *Onder **functionaliteit** wordt een deel van het eindproduct/project verstaan wat tussentijds opgeleverd **kan** worden en business value levert.*
- *Onder **business value** wordt in dit onderzoek een functionaliteit verstaan die toegevoegde waarde oplevert voor de klant.*

PROJECT GEGEVENS

Naam project

Noteer in het onderstaande tekstveld de naam van het project.

ALGEMENE PROJECTKENMERKEN

1. Project budget

Geef aan binnen welke budgetcategorie dit project valt:

- ☐ € 0 – € 50.000
- ☐ € 50.000 – € 100.000
- ☐ € 100.000 – € 250.000
- ☐ € 250.000 – € 500.000
- ☐ € 500.000 – € 1.000.000
- ☐ € 1.000.000 – meer €

2. Project-uren

Geef aan binnen welke geschatte uren categorie dit project valt:

- ☐ 0 – 500 uren
- ☐ 500 – 1.000 uren
- ☐ 1.000 – 1.500 uren
- ☐ 1.500 – 2.500 uren
- ☐ 2.500 – 5.000 uren
- ☐ 5.000 – meer uren

3. Cluster

Geef aan binnen welk cluster dit project valt:

- ☐ Algemeen
- ☐ Connectivity & Datacenters
- ☐ Connections
- ☐ Storage/X86/VMware
- ☐ NSK, z/OS & Oracle
- ☐ AIX/Linux/Websphere
- ☐ Windows/SQL/SharePoint
- ☐ IT-4-IT

4. Soort IT-Infrastructuur project

Geef aan wat voor soort IT-infrastructuur project dit betreft:

☐ Onderzoek
☐ Life Cycle Management
☐ Verbeteren bestaande dienstverlening
☐ Creëren nieuwe infrastructuur dienst
Let op: hiermee wordt bedoeld het creëren van nieuwe diensten binnen de Rabobank.
☐ Anders namelijk,

5. IT-Infrastructuur laag

Geef aan binnen welke infrastructurele laag dit project valt.
Let op: Vink 1 optie aan! De lagen onder de laag waarbinnen het project valt worden als dienst gezien om het project te realiseren.

☐ Beheer services/tooling/monitoring
☐ Eindgebruikers: apparaten/OS/Middleware
☐ Middleware/servers
☐ Operating systems/servers
☐ Databases
☐ Storage
☐ Servers
☐ Netwerken
☐ Data Center Floor

KENMERKEN AGILE-SCRUM TOEPASBAARHEID

6. Team Size

Met welk van onderstaande schaling komt het aantal teamleden binnen dit project het beste overeen:

☐ 0 - 3
☐ 3 - 10
☐ 10 - 30
☐ 30 - 100
☐ 100 – 300

7. Project complexiteit

Geef aan welk van onderstaande situaties de relatie tussen het probleem en de oplossing voor dit project vanuit een klantwens perspectief het beste omschrijft.

☐ ***Simple.***
De wens van de klant/opdrachtgever was vanaf de start van het project (na de PID fase) volledig duidelijk en daarmee ook de relatie tussen het probleem en welke oplossing er gebruikt moest worden. Bijvoorbeeld: LCM project het is duidelijk waarom iets vervangen moet worden en wat het doel is.

☐ ***Complicated.***
De wens van de klant/opdrachtgever was vanaf de start van dit project volledig duidelijk en daarmee ook de relatie tussen het probleem en de oplossing. Maar er waren meerdere oplossingen voor het probleem. Een afweging tussen deze mogelijkheden moest gemaakt worden.

☐ ***Complex.***
De wens van de klant/opdrachtgever was vanaf de start van het project deels duidelijk maar niet volledig. Waardoor wijzigingen/toevoegingen in requirements/scope van het project mogelijk waren. De klant had nog niet alle functionaliteiten van het volledige eindproduct voor ogen.

☐ ***Chaotic.***
De wens van de klant/opdrachtgever was vanaf de start van het project volledig onduidelijk en daarmee ook de relatie tussen het probleem en welke oplossing er gebruikt moest worden. En er moest met spoed naar een oplossing voor het probleem gezocht worden.

☐ ***Disorder.***
Dit project kan niet in zijn geheel geschaald worden in één van bovenstaande categorieën. Beschrijf in onderstaand tekstveld waarom dit project niet in zijn geheel in één van bovenstaande categorieën valt te schalen.
Let op: tekstveld alleen invullen bij keuze disorder!

8. Type stakeholder

Geef aan welk van onderstaande opties het type stakeholder situatie van dit project het beste omschrijft.

☐ ***Single organisation***
Het project is volledig binnen Rabobank Nederland uitgevoerd. De klant/opdrachtgever is ook intern en bij dit project waren geen externe partners/leveranciers of onderaannemers betrokken.

☐ ***Single organization / multiple BU's***
Het project is uitgevoerd binnen Rabobank Nederland maar de afdelingen/domeinen die bij het Project betrokken zijn functioneren ieder als een autonome organisatie (met eigen doelstellingen).

☐ ***Multiple organisations***
Het project is uitgevoerd in samenwerking met externe partners/leveranciers en/of onderaannemers.

☐ ***Anders namelijk,***

9. Type functionaliteit

Bedenk in hoeverre dit project op te delen is in deelfunctionaliteiten welke business value leveren aan de klant/opdrachtgever. (Ook al heb je dit project niet in verschillende deelfunctionaliteiten opgeleverd). Licht je keuze toe in het tekstveld!

Voorbeeld:
Beeld je in dat je eindproduct een pizza ananas is. De ingrediënten worden als functionaliteit gezien. Doordat je met ieder ingrediënt wat je toevoegt dichter bij het eindproduct komt (pizza ananas) voeg je (business) value toe in de ogen van de klant.

Let op: voor definitie functionaliteit & business value zie voorblad questionnaire!

☐ ***Één functionaliteit***
Het eindproduct voor dit project is niet/heel moeizaam op te delen in verschillende functionaliteiten die op zichzelf business value aan de klant/opdrachtgever leveren. Voor dit project had ik geen keuze om tussentijds een werkend gedeelte van de volledige oplossing op te leveren. De oplossing kon enkel in 1x geïmplementeerd worden om business value aan de klant te leveren.

☐ ***Meerdere deelfunctionaliteiten***
Het eindproduct voor dit project is op te delen in verschillende functionaliteiten die op zichzelf business value aan de klant/opdrachtgever leveren. Voor dit project had ik wel een keuze om tussentijds een werkend gedeelte van de volledige oplossing op te leveren. De oplossing had ik in meerdere delen, welke business value leveren, op kunnen leveren aan de klant.

Licht hieronder je keuze s.v.p. toe!

Dankjewel voor het invullen van deze 9 vragen!

P.S. Heb je meer dan 1 project, vul dan s.v.p. dit onderzoek ook voor je andere projecten uit 2014 in.

- einde -

Appendix V: Transcription and Coding

This appendix provides the results of the recorded focus group session and the open comment fields of the survey. Codes from both are sorted in the following categories: *functional product increments can be delivered* and *functional product increments can't be delivered*. The results are shown per type of IT Infrastructure project and used in the conclusion of this research.

Life Cycle Management projects – functional product increments can't be delivered

From the answers given by participants to the question: "*What is the reason that functional increments of the final product can't be delivered in this project*" a total of **31 instances** were identified. Preserving functionality is mentioned 15 times by the participants followed by continuity with 11 instances.

Code	Nr. Instances
Preserving functionality	15
Continuity	11
Project formulation	4
Long throughput time	1

Preserving existing functionality is the most mentioned reason why functional product increments of the final project can't be delivered within Life Cycle Management projects. The importance of continuity of these existing services makes that the final product has to be delivered at once.

Another interesting observation mentioned by some respondents is the availability of new functionality during almost every upgrade. But they mentioned that these projects are formulated in a way that the implementations of these new functionalities were out of the scope of the project.

One participant mentioned the following: "*Actually we are just executors, it's predetermined whether additional functionality should be offered or not*"

Another respondent said the following about the project formulation: *"This project was only aimed at preserving existing functionality, it was just a migration. But possibilities are created to offer new functionality. But this will be executed by the line-organisation."*

Life Cycle Management projects – functional product increments can be delivered

From the answers given by participants to the question: "*What is the reason that functional increments of the final product can be delivered in this project*" a total of **2 instances** were identified. Important to note is that these 2 instances are just from one respondent.

Code	Nr. Instances
Technical possibilities	1
Additional functionality	1

Only one respondent mentioned that there was a possibility to deliver functional product increments of the final product within a sprint up to 4 weeks. He mentioned the following reasons for this: "*It's technically possible and the additional functionality was included in the scope of the project*". A final note is that the existing functionality had to be delivered at once for continuity reasons.

Improving Existing Infrastructure Services – functional product increments can't be delivered

From the answers given by participants to the question: "*What is the reason that functional increments of the final product can't be delivered in this project*" a total of **9 instances** were identified.

Code	Nr. Instances
One service/functionality	6
Long throughput time	4

Adding a single functionality to existing services seems to be an important reason why functional increments of the final product can't be delivered. Another interesting reason mentioned by the respondents is that delivering a single functionality takes longer than a sprint up to 4 weeks.

Improving Existing Infrastructure Services – functional product increments can be delivered

From the answers given by participants to the question: "*What is the reason that functional increments of the final product can be delivered in this project*" a total of **8 instances** were identified.

Code	Nr. Instances
Expansion tooling	4
Expansion automation	3
Replacement infra service	1

Most of the respondents mentioned that functional increments of the final product can be delivered within a sprint up to 4 weeks within product aimed the expansion of tooling or automation. One respondent mentioned that he delivered functional increments of the final product during a project aimed at the replacement of an existing infrastructure service. Important to note is that he had organisational support to move over in phases from the old to the new system.

Creating New Infrastructure Services - functional product increments can't be delivered

From the answers given by participants to the question: "*What is the reason that functional increments of the final product can't be delivered in this project*" a total of **6 instances** were identified.

Code	Nr. Instances
Long throughput time	6

Only one reason could be identified why functional increments of the final product couldn't be delivered. The respondents mentioned that the delivery of a single functionality takes much longer than the sprint time of 4 weeks.

Creating New Infrastructure Services - functional product increments can be delivered

From the answers given by participants to the question: "*What is the reason that functional increments of the final product can be delivered in this project*" a total of **10 instances** were identified.

Code	Nr. Instances
Automation tooling	3
Assign authorisation	3
Configuration tooling	2
New Infra Stack	2

The respondents mentioned that within projects regarding the configuration and development of infrastructure tooling functional increments of the final product could be delivered within a sprint up to 4 weeks. Within projects regarding the assignment of multiple authorisations and/or authentications this is also possible. Finally there is a possibility to deliver multiple function increments of a new infrastructure stack. A side note here is that delivering the basic infrastructure takes longer than a sprint up to 4 weeks.

Research Projects - functional product increments can't be delivered

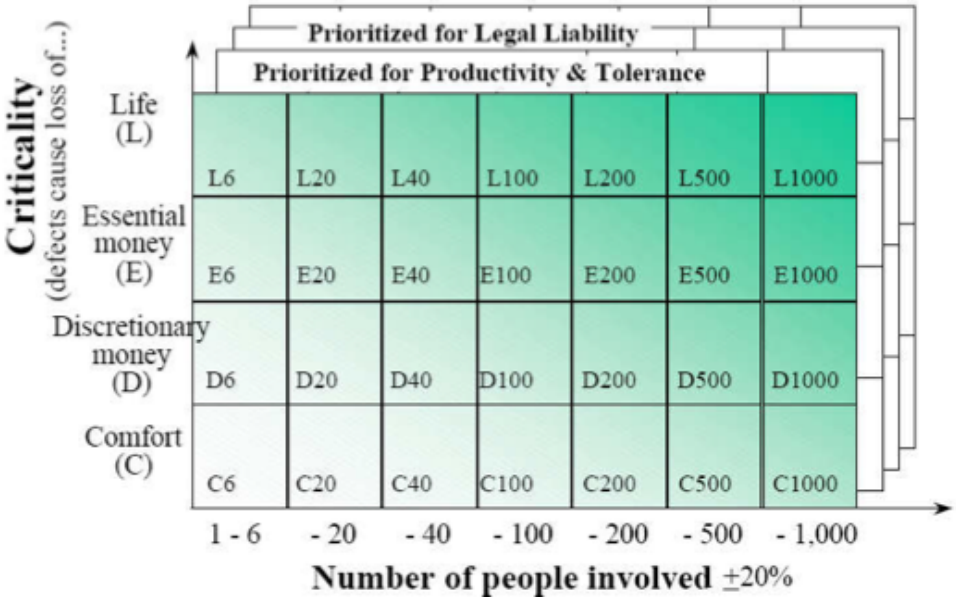
From the answers given by participants to the question: "*What is the reason that functional increments of the final product can't be delivered in this project*" a total of **6 instances** were identified.

Code	Nr. Instances
Advisory Report	6

Functional increments of the final product can't be delivered because it's an advisory report. But some respondents mentioned that for Research Projects in the context of a Proof of Concept there could be a possibility to deliver functional increments of the final product. Unfortunately none of the validated Research Projects in 2014 was a Proof of Concept.

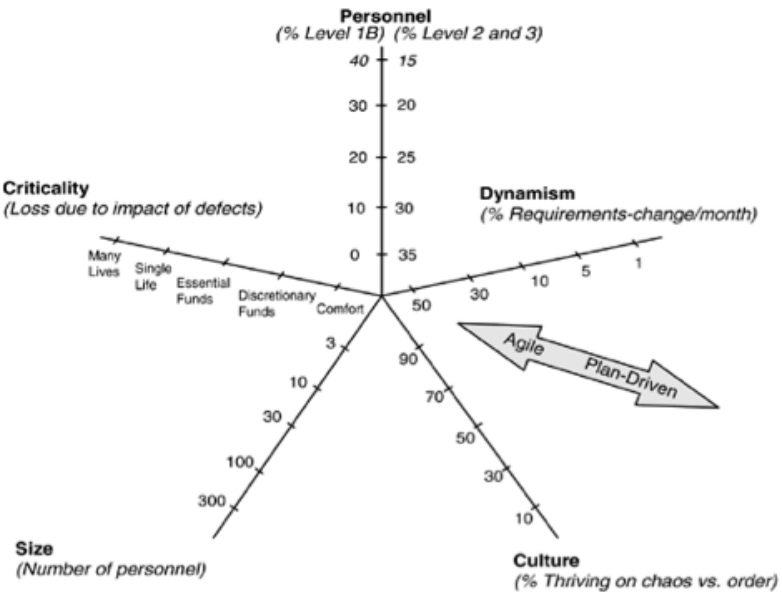
APPENDIX FIGURES

Figure III: Cockburn's Criticality and team size factors



Explanation: Cockburn argues that an Agile approach is applicable for projects that score close to the bottom left corner of the model, for example the score at C6. A traditional approach is recommended for projects that score further from the bottom left corner, for example the score at E200.

Figure IV: Radar Chart (Boehm & Turner)



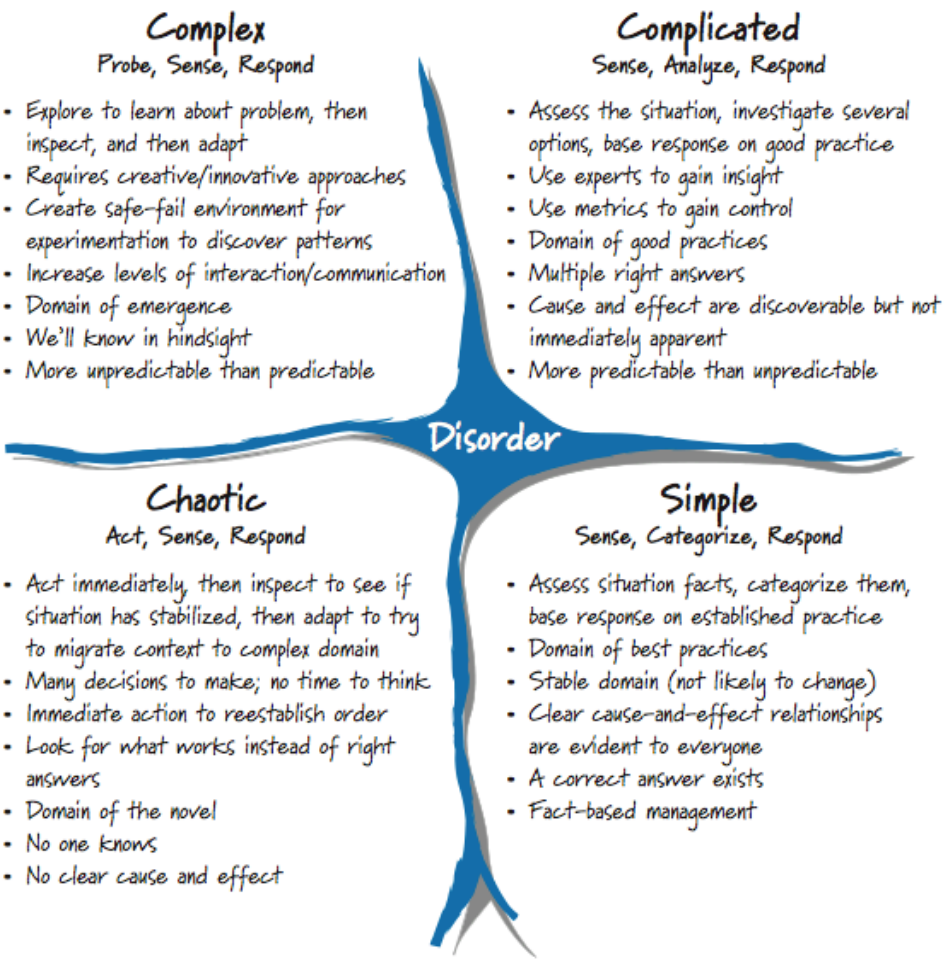
Explanation: this model points out that an Agile approach is recommended for projects that score in the centre of the model on the 5 characteristics. If not, a traditional project management method is recommended.

Figure V: Chin Model

	Multiple, External Stakeholders	Multiple, Internal Stakeholders	Single Organization
Operational Projects	Classic	Classic	Classic
Product/Process Development Projects	Classic/Agile	Classic/Agile	Agile
Technology/Platform Development Projects	Classic/Agile	Agile	Agile

Explanation: With his model Chin recommends an Agile approach for projects in the categories product/process/technology/platform development and they need to be independent of internal and external parties that pursue their own goals. Within operational projects a traditional project management method is recommended.

Figure VI: Cynefin Framework (Snowden & Boone)



Explanation: Snowden & Boone recommend an Agile approach for projects that fall into the complex domain. A traditional method is recommended for projects that are in the chaotic, complicated and simple domain.