Agile enterprise information system implementation

evaluating the applicability of agile practices in enterprise information system implementation projects : a case study

van Eldijk, J.

Award date:
2017

Agile Enterprise Information System Implementation

Evaluating the applicability of Agile Practices in Enterprise Information System Implementation projects: A Case Study

By Jochem van Eldijk
BSc. Industrial Engineering and Management Sciences (TU/e - 2015)
2/12/2017

In partial fulfillment of the requirements for the degree of

Master of Science

In Business information systems

Supervisors:

Dr. O. Türetken	TU/e, information system
J. Stegeman	Company

Voting members of assessment committee:

Dr. O. Türetken	TU/e, information system
J. Stegeman	Company
Dr. Dirk Fahland	TU/e, Architecture of Information Systems

Preface

This document is the result of my graduation project I carried out at the Company[1], in conjunction with my school, Technical University of Eindhoven (TU/e). The graduation project is part of the last phase of my master Business information systems with a specialization in information systems. During my graduation project I really learned a lot and enjoyed acquiring knowledge about implementations of enterprise information systems and agile software development methods. Most of all I enjoyed the pilot project in which we evaluated the implementation method.

First and foremost I would like to thank Joris for being my tutor at the company, without his enthusiasm and commitment this project would not have been possible. Always quick to supply me with the information I needed and provide guidance on how to get things done within the organization. His help with applying the theory on agile development within the context of the company was invaluable.

Second, I would like to thank Oktay for being my first supervisor for this project, even though he already supervised more master students then he normally would have done. He was a great help with determining the content and structure of this report and also offered valuable feedback. All of which have greatly improved the quality and readability of this report.

As third, I would like to thank Dirk for being a voting member on my assessment committee and for his feedback on this report.

Finally, I would also like to thank: all of the people who participated in the pilot project, my brother for letting me stay in his house and my girlfriend for her endless support.

[1] Due to confidentiality reasons the name of the company, in which this research is conducted is not mentioned in this report

Abstract

A large amount of enterprise information system (EIS) implementations are labeled as unsuccessful, exceeded their planned budgets and experienced schedule overruns. These results can to a large extend be contributed to the implementation method that is used.

In the late 1990's a new type of development method emerged, which has been reported to offer great benefits for small software development projects. These methods are known as agile development methods and provide concrete practices that can be applied within a project. However, remains to be seen if these practices can also be applied in large EIS implementation projects and if they offer the same benefits.

This study investigates which agile practices can be applied within an EIS implementation and what their affects are. This is done by taking the following steps: First agile practices that used in scrum and extreme programming (XP) are selected by a group of implementation experts in a set of semi structured interview. Those practices are placed within a software development life cycle, which is evaluated during a focus group. To validate the final agile implementation method, it is applied in a case study. At the end of the case study semi-structured interviews and a survey are used to gather results.

The interview results showed that the "project chart", "planning game", "product backlog", "sprint", "daily meeting", "sprint review", "task board", "retrospective", "user story", "story points" and "velocity" practices can be applied within an EIS implementation and that applying these practices resulted in an increased: productivity, sense of responsibility, motivation, usability, customer satisfaction, flexibility, transparency and a decrease in the number of bugs found.

The survey showed that the level of agility of the case study was close to the world wide benchmark, indicating that the agile way of working was successfully adapted within an EIS implementation project. It can therefore be concluded that agile practices can be applied in EIS projects, which can improve the projects performance.

Table of content

List of figures

List of tables

List of abbreviations

EIS	Enterprise information system
ERP	Enterprise resource planning (system)
SDLC	Software development life cycle
AIM	Agile implementation method
US	User story
AT	Acceptance test
N/A	no answer

1 Introduction

Today's businesses have to deal with an ever increasing competition, changing markets and higher costumer demands. This requires companies to lower their cost, increase their throughput, provide more service, improve their quality, and efficiently coordinate global demand, supply, and production (Brit, 2007; Dahlman & Edmund, 2007; Shankarnarayanan, 2000). To accomplish these objectives many companies are tuning to enterprise wide software solutions called enterprise information system (EIS) or enterprise resource planning (ERP) systems. EIS systems are built around industry best practices (Stender, 2002) and provide a unified enterprise view of the business that encompasses all functions and departments. To do so, it makes use of an enterprise database where all business transactions are entered, recorded, processed, monitored, and reported. This enables interdepartmental communication & coordination, increased responsiveness and communication to all stakeholders (Dillon, 1999). This is especially relevant as "the success of a company increasingly depends on timely information (internal and external) being available to the right person at the right time for crucial managerial decision-making," quoted from (Chen, Andrew, Goes, Gupta, & Marsden, 2006).

With the above mentioned features, EIS appear to be a dream come true for managers who have struggled with incompatible information systems and inconsistent operating practices. To them the promise of a ''off-the-shelf'' solution to the problem of business integration is un-resistible. However, it's not all fun and games. According to a recent survey in practice (2015), approximately 58-percent of the EIS implementation project exceeded their planned budgets and 65- percent experienced schedule overruns. Post implementation, 53-percent of organizations achieved less than 50-percent of the measurable benefits they anticipated from new EIS software. In the last few years the overall satisfaction with selected software has continued to decrease and more companies are defining their projects as failures (21% in 2015 compared to 16% in 2014) (Panorama Consulting Solutions, 2015). The success, costs and risks of an EIS implementation are all related to the implementation method that is chosen by the organization (Sindos, 2001). It is therefore worthwhile to take a closer look at the EIS implementation methods that are available.

The most common implementation method that is used today is the traditional waterfall approach (Gartner, 2016). The waterfall approach follows a linear process, consisting of a series of basic sequential steps, each of which needs to be validated before moving to the next (Ruël, Bondarouk, & Smink, 2012). One reason of why this approach is so popular is that, ERP software vendors typically prescribe an implementation methodology with their software. For over 20 years these software packages have been delivered with traditional waterfall implementation methodologies; therefore, the system implementation communities have also adopted their implementation models to be in alignment with the ERP vendors (Fair, 2012). Another reason might be that the waterfall approach offers a relative easy way to manage and control the development process by a having a strict and orderly sequence of development steps (Zughoul, 2016). This is especially the important for EIS implementations, which are large project that have big impact on a company. An average project lasting 14.3 months and costing 4.5 million US dollars (Panorama Consulting Solutions, 2015).

An alternative to the waterfall approach emerged in the software development domain since the late 1990's, called the agile approach. Where waterfall approaches focus on prediction and controllability, agile methods focus on change and being able to correctly handle that change. Agile is also about team structures and attitudes that improve communication. It emphasizes rapid delivery

of operational software and de-emphasizes the importance of intermediate work products. It adopts the customer as a part of the development team and works to eliminate the “us and them” attitude. Agile methods recognizes that planning in an uncertain world has its limits and that a project plan must be flexible.

Of course these differences are no reason for choosing one method over the other. However, agile software development methods also offer a wide range of project related benefits. In a survey with 2229 respondents conducted by Doyle, Williams, Cohn, and Rubin (2014) the following benefits were listed when comparing agile development with previous approaches:

- The team has produced higher quality products
- The team has been more productive
- The team has had higher morale
- Customers have been more satisfied with the functionality of the products
- Customers have been more satisfied with the usability of the products
- Our business has recognized greater economic value
- Delivered functionality to users more quickly and/or more often

It should, however, be noted that agile development methods originally have been developed for small software development projects. In a workshop held by (Lindvall et al., 2002) that included 18 agile experts like: Alistair Cockburn and Kent Beck, it was agreed that:

- Agile with face-to-face communication breaks down and becomes more difficult and complex past 20-40 people
- Past 20-40 people some kind of scale-up strategies must be applied

It therefore remains to be seen if the benefits reported in smaller software development project can also be achieved by large EIS implementation projects.

In a study previously performed by the authors of this report, it was found that there are some studies that adapted some of the practices used in agile software development to fit EIS implementations. Although, most of these works show the benefits and the applicability of their solution trough a good line of argumentation, there is a lack of studies that empirically evaluate these claims (Eldijk, 2016). On top of that, the proposed methods only use a small set of the agile practices available in the literature.

Therefore, this study aims to develop a new agile implementation method (AIM) for EIS implementations by selecting and incorporating a broad range of agile principles. The goal of the AIM is to increase the team’s productivity and morale, the products quality and usability and customer’s satisfaction. To assess if AIM achieves these goals, a case study is conducted. By creating this method and validating it in practice, this study aims to address the following research questions:

- Which agile practices are applicable in an EIS implementation?
- What are the effects of applying these agile practices in EIS implementations?

A literature review was conducted to review the current state of existing academic literature, to identify the research gap, and to demonstrate the (potential) contribution of this research. A new EIS implementation method based on practices of scrum and XP is introduced and validated in a case

study. This study, therefore, extends the findings on applying agile practices in EIS implementation projects. The results can also serve as input to compare existing EIS implementation methods. This will enable practitioners to select a suitable implementation method and academics to extend the existing methods or to develop a method themselves.

The rest of the report is structured as follows. First, the background of agile practices and related work on EIS and implementation methods is given. This is followed by the research design of this study and the development of AIM is described. After this description, AIM is described, followed by the case study that evaluates the effects of applying AIM in practice. Finally, in section 8, the conclusions, and the limitations of this study, alongside directions for future research are discussed.

2 Background and Related work

This section provides the background on the tools, techniques and frameworks that were used in the development of AIM, and provides an overview of related work. First, the definition that this paper uses for EIS systems is given. Next, the traditional EIS implementation method, namely the waterfall, is discussed. This is followed by the discussion on the values and principles that form the basis for the agile software development methods. Next, a list of agile practices that apply the agile principles is provided and few agile software development methods are described. This section ends with an overview of the agile EIS implementation methods that are available in the existing literature.

2.1 Enterprise information systems

Over the last 20 years, several business software systems (such as the CRM - Customer Relationship Management, SCM – Supply Chain Management) have emerged, which are collectively named under the term enterprise information system (EIS). This report will use the EIS definition, as formulated by Klaus et al. (2000), who specify that an EIS system shows at least the following characteristics:

- It is based on an underlying integrated database that stores master and transactional data in a consistent way and with controlled redundancy.
- It provides business solutions, which support the core processes of the business and administrative functionality.
- It supports a range of typical business functions that all have a common user interface.
- It is configurable software based on best practice models.
- It includes various solutions for user administration, database configuration, system monitoring, or performance measurement.

2.2 Enterprise information system implementations

EIS implementations are different from normal software development projects in multiple ways. Where software development projects normally start in a Greenfield situation in which all features still need to be developed (Dalling, 2017), EIS implementations already have a working system that only needs to be configured to the specific context. When configuring a system, the options of the implementer are limited to the options that the EIS offers (Davenport, 1998), whereas a software developer can write any code s/he wants within a given programming language. This flexibility comes at the price of having to come up with a new solution, which takes up more time than making a selection from a set of options (Bebbington, 2014).

Based on this information one might conclude that EIS implementations are simpler than software developments, but that would be a mistake. As discussed in the previous subsection, EIS systems usually cover multiple business processes across different organizational departments. This makes these type of projects take up large amounts of time and resources (Panorama Consulting Solutions, 2015). An average software development project only takes up a fraction of that amount (Flackett, 2015).

It is therefore not surprising that the traditional EIS implementations life cycle follows a formal management procedures that offering a lot of control and emphasize in depth planning and documentation. The phases in this traditional approach are executed sequentially and only once for each system. Supporters of this approach emphasize that changes made in earlier stages of a project are less expensive than those made in the later stages (Alleman, 2002). This linear sequential model is often called waterfall and is a model derived from engineering process, which contains the following model contains six discrete phases: initial investigation, requirements definition, system and software design, development and testing, implementation, and operation and support (Zughoul, 2016). Figure 1 gives a visual representation of this process.

Figure 1: waterfall model

In (Alleman, 2002) it is stated that the wide spread use waterfall method is understandable in many EIS deployment cases, since most of those companies have a tradition in engineering. This tradition makes them think in terms of waterfall development processes. However, Alleman (2002) also alerts that "the use of predictive strategies in this environment is inappropriate as well as ineffective since they do not address the emergent and sometimes chaotic behaviors of the market place, the stakeholders, and the vendors' offerings." This chaotic behavior combined with the amount of time that elapses between setting software requirement and software delivery, makes it likely that the delivered software will not meet stakeholder expectations (Wiegers & Beatty, 2013).

2.3 Agile principles and values

In 2001, Kent Beck and 16 other noted software developers wrote the "Manifesto for Agile Software Development." The goal of this manifesto was to overcome the weaknesses of traditional software engineering (the waterfall method). Their manifesto stated:

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

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

That is, while there is value in the items on the right, we value the items on the left more."(Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J. and

Thomas, 2001). In their manifesto they also translated these values into the following 12 principles, which should function as a guideline for agile methods:

1. Customer satisfaction is the highest priority and is achieved by early and continuous delivery of valuable software.
2. Changing requirements are welcomed even late in development.
3. Working software is delivered in short cycles. These cycles can take a couple of weeks to a couple of months. But there is a preference for the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Empower and support developers and trust them to get the job done.
6. Conveying information is preferably done face-to-face.
7. Working software is the primary measure of progress.
8. The sponsors, developers, and users should be able to maintain a constant development pace indefinitely.
9. Continuous attention to technical excellence and good design.
10. Simplicity—only doing wat 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.

Not every agile method applies all of the 12 principles with equal weight. Some methods ignore or downplay the importance of one or more of the principles. However, all agile methods do share the "agile spirit" that these principles define (Pressman, 2009).

2.4 Agile practices

The principles mentioned above are too vague to be used in practice. Thankfully, there are a large number of agile practices that provide concrete guidance to any one that wants to adopt (a part of) the agile values. These practices cover a broad spectrum of project areas like requirements, design, modeling, coding, testing, planning and risk management. Table 1 provides an overview of a large number of the agile practices relevant to this work. The first column of this table provides the name of the practice and the second column provides the description. This list and descriptions are based on the works of (Abrahamsson, Salo, Ronkainen, & Warsta, 2002; Agile Alliance, 2015; Cohn, 2004, 2006; Pressman, 2009).

Table 1: list of agile practices

Practice	Description
Sprints / small releases	Development cycles of a fixed length, at the end of which a demonstrable piece of functionality is delivered.
Planning game / Effort estimation	programmers estimate the effort needed for the implementation of user stories and the customer then decides about the scope and timing of releases
Planning Poker	Estimation approach in which each team member places an effort estimate card. Based on those estimates a discussion is held to come to a consensus

Time-box	Fixed period of time during which a team works steadily towards completion of some goal
Daily meeting	Daily meeting in which each team members briefly describes any "completed" contributions, any obstacles that stand in their way and their goals for that day.
Sprint backlog / task board	Board that displays the status of tasks as either: “to do”, “in progress” or “done”.
Iteration/sprint retrospective	Meeting in which the team reflects on the most significant events that have occurred in that iteration, and take decisions aiming at remediation or improvement
Story Mapping	Way of ordering user stories along a horizontal time axis and vertical axis that order them on time and necessity dimensions respectively (Pattion, 2008; Patton, 2005)
Burndown chart	A burndown chart is a graph which shows the quantity of work remaining and the time elapsed since the start of the project
Velocity	velocity of a team is the amount of user stories that were completed during that sprint expressed in the units used for effort estimates
Project Chartering	project charter is a high-level summary of the project's key success factors that are displayed on one wall of the team room as a flipchart-sized sheet of paper
User stories	User stories are requirement that are written on a physical card. These requirements are a functional piece of the product that provides value to the customer and can be implemented independently from each other.
Frequent Releases	Agile team frequently releases its product into the hands of end users in order to gain feedback, weather that is positive or negative
acceptance Testing / definition of done	An acceptance test is a description of the behavior of a software product, expressed as an usage scenario. Executing the acceptance test generally result in either a pass or fail
Collective code Ownership	Convention that "every" team member is not only allowed, but indulged, to make changes to "any" code file as they see fit.
On-site customer / product owner	A real system user must sit with the team, available to answer questions, resolve disputes and set small scale priorities.
Pair programming	Practice in which two programmers share a single workstation. The programmer at the keyboard focuses on writing code while the other focuses more on overall direction.
Refactoring	Process of changing a software system in such a way that it does not alter the

	external behavior, yet improves the internal structure.
Sign up	Practice in which team members choose which tasks to work on.
Product backlog	list of features or technical tasks which at a given moment, are known to be necessary and sufficient to complete a project or a release
Common workspace	dedicated space used by the whole team for the duration of the project, set apart from other groups' activities
Automated testing	Short program fragment written and maintained by the developers, which exercises some narrow part of the product's source code and checks the results automatically.
Test driven development	Style of development that starts with writing a unit tests and then writing just enough code to pass each test.
Continuous integration	Developer's working copies are synchronized with a shared mainline several times a day.
40-hour week	Type of contract negotiated between the team and their management which specifies a work pace that they would be able to sustain indefinitely. Overtime, other than on an exceptional basis, is seen as detrimental to productivity rather than enhancing it.
Simple design	Emphasis is on designing the simplest possible solution that is implementable at the moment.
Coding standards	Coding rules exist and are followed by the programmers. Communication through the code should be emphasized.
Story point	measure used to indicate the size of a user story relative to one another

2.5 Agile methods

Agile methods support a broad range of the software development life cycle (SDLC) (Abrahamsson et al., 2002). The SDLC is the set of activities conducted in a set of phases, which each project goes through when realizing a software system (Hoffer, 2011). Any system development life cycle should encompass project definition, system study, design, programming, installation, and post-implementation stages. Some agile method cover the full range of the SDLC (like Rational unified process and Dynamic systems development method), but most range from requirements specification to system test. Some focus on the practices and activities that should be applied (e.g. Extreme programming) while others focus on how the project should be managed (e.g. Scrum). A complete overview of the phases and aspects a specific agile methods covers are shown in Figure 2. Appendix A gives a brief introduction into the two well-known methods: scrum and extreme programming.

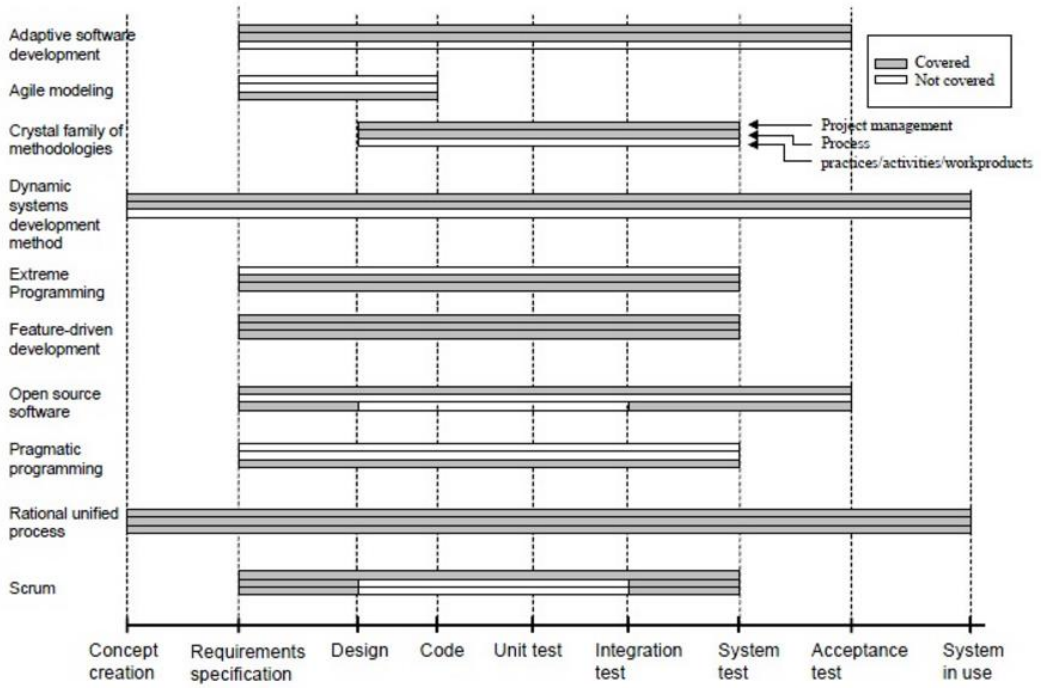


Figure 2: parts of SDLC that different agile methods support (Abrahamson P, 2002)

In the last decade, more and more companies are transitioning away from the traditional waterfall approach in favor of the more flexible agile methods (Dingsoyr, Nerur, Balijepally, & Moe, 2012). However, research suggests that many organizations do not fully adapt all aspects of agile, but instead integrate both traditional and agile practices into a customized method (Cram & Marabelli, 2015). A project may, for example, use a traditional practice such as up-front planning alongside an agile practice such as pair programming. By tailoring a development approach to match the characteristics of a process, customer, or organization, the performance of the development can be enhanced (Cram & Marabelli, 2015; Fitzgerald, Hartnett, & Conboy, 2006).

2.6 Agile deployment frameworks

It should be evident that the transition from a traditional waterfall approach to agile way of working does not happen overnight. It is a process that requires hard work, strict discipline and continues learning and improvement (Schatz & Abdelshafi, 2005). It has also noted that an agile approach requires a different mindset, process, people, environment and tools for its successful implementation (Qumer & Henderson-Sellers, 2008). This section, therefore, discusses some of the frameworks that are proposed in academic literature. First two maturity models are covered, followed by an agile deployment framework.

Maturity models can be a useful guide for organizations that want to adapt a set of practices in the organization. The basic purpose of maturity models is to outline the stages an organization goes through when transforming their way of working. Ozcan Top & Demirors (2013) present a case study based comparison of nine agile maturity models, out of which the model proposed by Sidky (Sidky Agile Measurement Index [SAMI]) scored the highest. However, SAMI by itself does not guide organizations adopting agile practices. Thankfully, the authors of SAMI (Sidky, Arthur, & Bohner, 2007) also provide a second component in their Agile Adoption Framework: the 4-Stage process. This

4 stage process uses SAMI to guide organizations by identifying the agile practices that are most suitable for their environment. The four stages are:

- Stage 1: Identification of Discontinuing Factors
- Stage 2: Project Level Assessment
- Stage 3: Organizational Readiness Assessment
- Stage 4: Reconciliation

Stage 1 is a pre-assessment phase in which the decision is made to start the initiative. Therefore, the objective of this first stage is to assess whether the organization is capable of embarking on the journey of transitioning to agility. Stage 2 starts once a Go decision is made from Stage 1. The main objective of this stage is to identify the highest level of agility that a project can achieve. This assessment process is accomplished by utilizing the assessment indicators identified within SAMI. The objective of Stage 3 is to determine the extent to which the organization is ready to support the adoption of the project's target agile level specified in stage 2. The objective of stage 4 is to reconcile any difference that may exist between the two levels set in stage 2 and 3. During this stage, the differences between the practices the project wants to adopt (i.e. the project's target agile level) and the practices the organization can adopt (i.e. organization's readiness level) are resolved. This stages results in determining the set of agile practices that are most suitable for the organization to adopt.

Another maturity model is described in (Qumer & Henderson-Sellers, 2008), which is called the Agile Adoption and Improvement Model (AAIM). That can guide a software development organization to adopt and improve agile practices for a specific situation or project. Their model shown in Figure 3, which contains 6 AAIM levels (AAIML) that are grouped in 3 blocks. These blocks range from basic at the bottom, to advance at the top. Each level establishes agile practices that a company can implement. Once all practices of a level are implemented, a company can move up one level, thereby increasing the degree in which agile is applied within an organization.

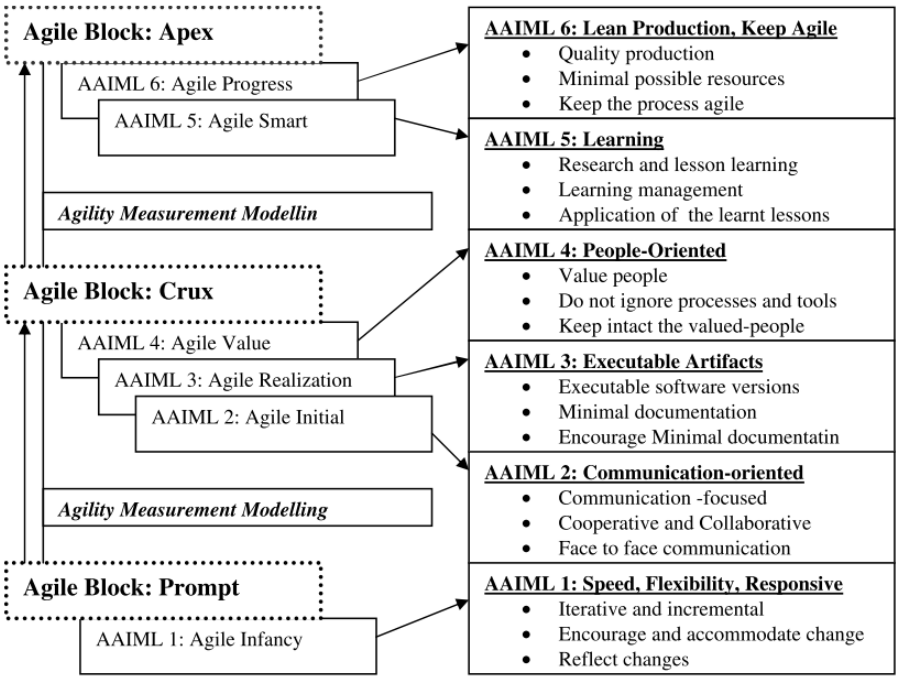


Figure 3: Agile adoption and improvement model (Qumer & Henderson-Sellers, 2008).

Rather than prescribing a predefined set of agile practices, Sureshchandra and Shrinivasavadhani (2008) propose to ask the development team, after a training on agile, to choose agile practices that will solve the problems they faced with their old methods. At the start, the team picks only user stories, large iterations, time-boxing and daily standup meetings from the set of agile practices. As the project evolves, the problems are encountered with regards to planning, analysis and testing. This is solved by adopting more agile practices, such as test driven development, pair programming and team effort estimations. This helps team members to immediately see the added value of these practices, which in turn increases the adaptation rates. Based on their experience, Sureshchandra and Shrinivasavadhani conclude that teams may not be ready to start with all the practices and that even simple practices, like standup meetings, also require guidance in the early stages of a project. They also note that changing the mindset of project managers takes some time and recommend them to talk to other managers who have already practiced agile.

2.7 Agile practices in Enterprise Information System Implementations

According to (Alleman, 2002) an agile EIS process should include tree major attributes. It should have an incremental, iterative and evolutionary delivery of system functionality. The development process itself should be modular and lean, meaning that components of the process should come and go depending on the project needs. Finally, the process should contain feedback loops used to improve the next iteration. Next to these attributes, Alleman (2002) proposes an options-approach for cost estimation and feature prioritization. Finally, an adaptation of 12 agile principles is provided and together they form a foundation for managing EIS implementation projects in an agile manner.

Cram and Marabelli (2015) performed a longitudinal case study on a company that applies a hybrid agile traditional approach. They compare knowledge management practices applied in the first launch of the CRM system with the relaunch where the hybrid method was applied. They report that the documentation and training remains traditional. Requirement management, project learning and knowledge repositories became agile, while team composition, trust & care, and competence management gained a hybrid form. Comparing the two implementations, the hybrid was viewed as more successful by the interviewees, who perceived the introduction of agile techniques as a key factor driving improved system performance.

Fetouh et al. (2011) proposes another hybrid method, which is a combination of big-bang and agile approaches. This approach, first, analyses the dependencies between different modules that need to be implemented. Once that is complete, the modules with the highest dependency are grouped together to be released within a single iteration. This ensures that every increment delivers a working part of the system. In addition, the number of interfaces that bridge the gap between legacy system and the new system is reduced. According to the authors, this will reduce costs, and will not take long or require large amounts of resources. It also allows going back to previous setup.

The work by (Sowan & Tahboub, 2015) considers the critical success factors of an EIS implementation in order to come up with an agile implementation methodology. This methodology starts with planning the whole project and dividing it into multiple iterations of 1 to 4 weeks. Each iteration starts with gathering and understanding requirements followed by a prototype, which is used during design and implementation. At the end of an iteration, validation and user training takes place. The user should be involved throughout all of the above mentioned phases and should meet regularly with the development team.

In (Kalaimani, 2016), the SAP ASAP 8 methodology is combined with scrum. The resulting approach contains the following phases: Opportunity, Project Preparation, Blueprint, Realization, Final Preparation, Go Live Support and Run SAP. During the *opportunity phase* the processes in broken down into simple tangible processes by discussing best practices and pre-configured scenarios. The *preparation phase* involves setting up the infrastructure, team, project goals, charter, and agree upon schedule, budget and risk baseline. *Blueprint* is the critical requirements gathering phase. This is where the sprint cycle start by showcasing pre-configured scenarios to come up with the product backlog. Each sprint consists of: analysis, realization, documentation and testing. In the realization phase you start implementing the backlog. Every sprint releases a set of features that are demonstrated and reviewed by the team and stakeholders, providing feedback for the next iteration. Documentation is kept to minimal as progress is measured based on the working product and/or feature delivered. Were necessary documentation is introduced in the code itself. Once all required releases are completed the project follows the basic ASAP method for Final Preparation, Go Live Support and Run SAP.

Stender, (2002) introduces an agile incremental implementation methodology (AI^{2}M). This method applies the incremental rollout, user Integration, automated testing, condign standards and minimal documentation practices used in XP. An iteration cycle has three phases. In *phase 1* overall business goal are identified. In *phase 2* a set of relevant processes to enable the business goal are identified, which are to be supported by the enterprise system. A planning game is held to determine the scope of the iteration. The process that is to be implemented is documented as lightweight as possible. Test cases are written before implementation and pair programming between user and system expert is aspired during an implementation. After completing a customization, automated tests verify the achieved result. When testing proves a desired level of quality is achieved, user tests verify the implementation from a user perspective. This phase finishes with a commitment to put the achieved results into production. *Phase 3* rolls out the system release of that iteration into the field. It should be noted that a cycle has a fixed time-box.

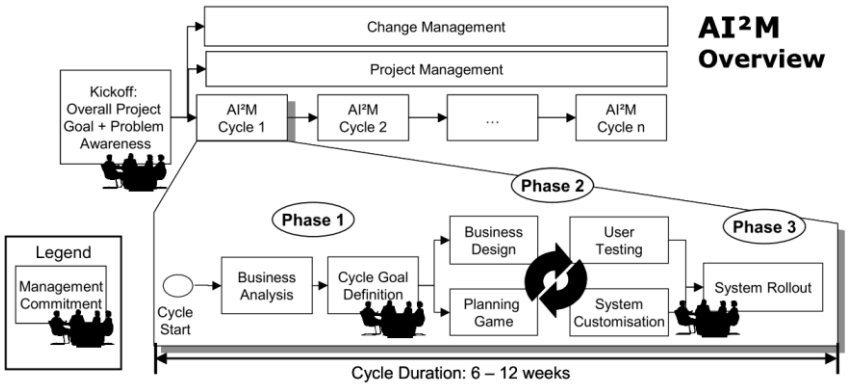


Figure 4: AI^{2}M Project Structure Overview source: (Stender, 2002)

Another incremental implementation method comes from (Finchman & Scoot, 1999). Their results-driven incremental (RDI) strategy has five key principles:

1. business results drive decision,
2. Implementation is divided into a series of non-overlapping increments that enable measurable business improvement.

3. Each increment implements everything required to produce desired results
4. Increments are sized so that each can be implemented in a short time (< three months)
5. Results of each increment are used as a basis to flesh out and adjust the plan for subsequent increments.

The methodology begins with a *Business Analysis* step that defines the contents of the first few business releases. Each business release needs to cover the following elements:

- The targeted business result.
- A list of software functionalities to be implemented.
- A list of complementary changes to organizational policies, procedures, measures, and structure
- A metric for measurement of business results.

The *Business Analysis* step identifies the first few releases at a high level, and defines the first release in detail. The details of subsequent releases are defined in "just in time" fashion following completion of the prior release. Following the Business Analysis step, the implementation team begins the process of completing the specified Business Releases. A business release is considered complete only when software functionalities are in actual production use, organizational changes have been made, and measurement of results has begun. Following the completion of each Business Release the project team reviews the results of the business analysis step for validity, the next Business Release is defined in detail, and a new Business Release is added to the queue such that a rolling set of Business Releases is maintained.

2.7.1 Why yet another implementation method is needed

Based on the list of available implementation methods provided in section 2.7, one might conclude that there are sufficient methods to choose from. However, all of those methods lack information on what to do during specific phases of an implementation project.

The first method proposed by Alleman (2002) only provides a set of agile practices, but does not tell in which stage of the project they should be applied. The method of Cram and Marabelli (2015) does not provide agile practices and only focuses on the knowledge management aspect of an implementation. The methods of Fetouh et al. (2011), Sowan & Tahboub (2015) and Finchman & Scoot (1999) only provide general descriptions of the different phases a project should go through, but does not provide agile practices that should be applied during those phases. The ASAP 8 methodology, proposed by Kalaimani (2016), focusses on the implementation of the complete EIS, but lacks details on how to handle the implementation / configuration of the individual processes. Finally, the AI^{2}M proposed by Stender (2002) does include agile practices and even provides a set of project roles. However, the method only uses XP practices and lacks practices on how to properly manage an implementation project.

There is, therefore, a need for yet another implementation method that does cover all of the aspects that the previously discussed methods lack. Thus, a new implementation method should satisfy the following requirements:

1. The method should include a set of agile practices that can be applied within an EIS implementation.

2. The method should have a SDLC that provides a sequence in which the agile practices should be applied.
3. The method should cover both project the management and the development aspect of an implementation.
4. The method should provide a set of roles, each of which responsible for carrying out certain practices.

3 Research Design

This research follows a design science paradigm, as it tries to solve a specific business problem by developing a software engineering artifact. In the case of this study, this artifact is a software development method that allows EIS to be implemented in an agile way. To build this artifact, this research follows the following steps: 1) define the problem and the objectives of the artifact, 2) design and develop the artifact, and 3) evaluate the artifact by applying it in a real life setting (Peffers, Tuunanen, & Gengler, 2006). Chapter 1 of this paper discusses the problems faced by EIS implementations and the objectives of AIM (step 1). The analyses of the problem and the design of the solution (step 2) are covered in chapter 4. Chapter 6 covers the evaluation of the AIM by applying it a case organization in order to assess its validity (step 3).

Figure 5: research steps.Figure 5 depicts a detailed view on the procedure that was followed in developing, applying, and evaluating the artifact proposed in this paper. The first phase involved the development of AIM. At the start of this phase, an initial set of agile practices were selected based on the literature review on agile values, principals, practices and methods (Eldijk, 2016). To increase the relevancy of the selected practices and to ensure the content validity, we performed a series of semi-structured interviews with a number of domain experts to refine and finalize the set of practices. Semi-structured interviews were used, because they help identify the meaning respondents attach to a phenomenon. Moreover, it lets researchers cross check their interpretations of some piece of information (Blumberg, Cooper, & Schindler, 2011).

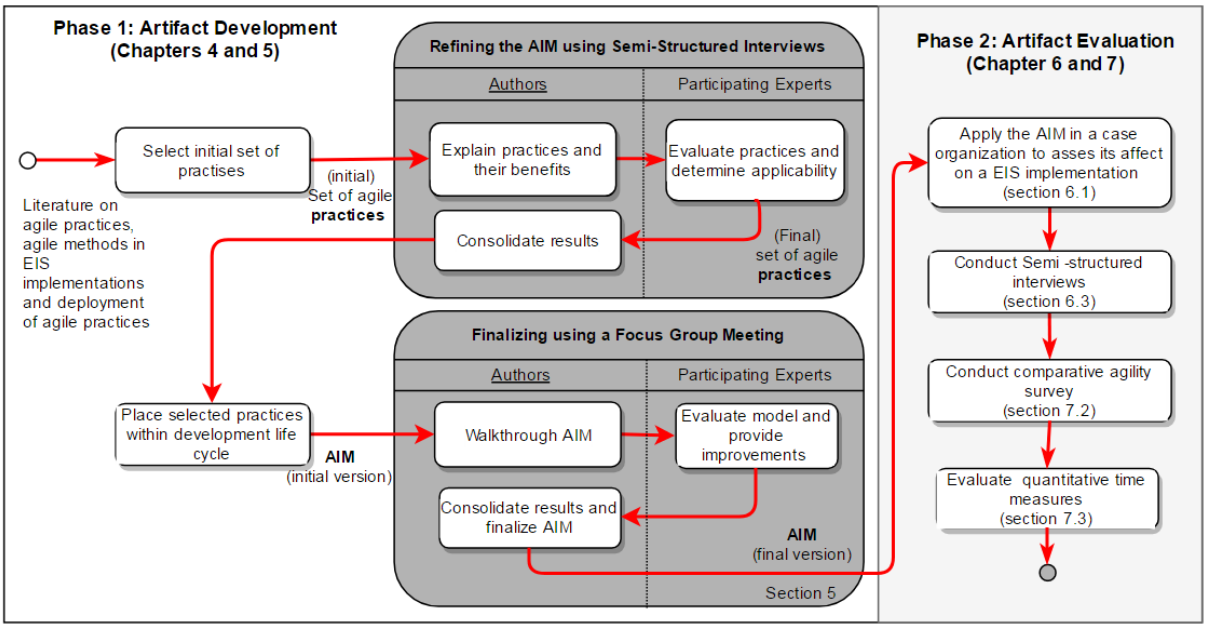


Figure 5: research steps.

Next, the selected practices were placed within a SDLC to create an initial version of AIM. To evaluate the applicability of AIM within an EIS implementation and to identify any potential risks, the method was presented to an expert panel during a focus group meeting. This method was chosen, because it allows information to be gathered from a group of senior experts in a cost and time effective way. It also helps in detecting the different views that people have and enables researchers to observe interactions between respondents (Blumberg et al., 2011).

In the second phase, the model was applied in a case study to assess the validity of the method in terms of the degree it can be applied, and to assess the effects it would potentially have on an EIS implementation process. The case study applied the AIM in a single implementation project at a client of the EIS vendor. The pilot included 4 members working for the EIS vendor and 4 members working for the company to which the EIS was implemented. The vendor team worked for 8 hours per week for a total of 8 weeks. Throughout the pilot, all participants recorded the amount of time they spent on each project activity. In section 7.3, these quantitative time measures are compared with time measures of two similar projects, in order to evaluate the pilot projects performance.

After the case study, structured interviews were held with five of the project participants: 3 members of the EIS vendor and 2 members of the client company. The interviews (where the participants indicated their perceptions on the gains) were used to infer the benefits of applying agile practices within an EIS implementation. A detailed description of how this case study was conducted and the findings of the semi-structured interviews are given in chapter 6. Those findings are evaluated in section 7.1.5.

In addition to the interviews, a survey was conducted during the artifact evaluation. This survey was developed by three of the four authors of (Doyle et al., 2014) and is used in section 7.1 to assess the agility level of the pilot project and compare it to an industry benchmarks.

4 Development of the Agile Implementation Method

This section discusses the steps taken in the development of the AIM. The method was developed in close cooperation with industry experts, following a series of structured steps. First, the initial set of agile practices was selected from the literature and validated in a series of semi-structured interviews with a number of domain experts. Second, the final set of practices was placed within a software development life cycle, which was reviewed by an expert panel during a focus group.

4.1 Selecting the initial set of practices

After an extensive review of the agile methods, we took the agile practices applied in Scrum (Agile Alliance, 2015; Cohn, 2004, 2006, 2010) and Extreme programming (XP) (Abrahamsson et al., 2002; Agile Alliance, 2015; Beck, 2004) as a starting point for our solution framework. Scrum and XP are selected mainly due to their comprehensive nature and popularity in software development (McLaughlin, 2016). Another reason was that Scrum focuses more on project management, while that of XP is on the activities and work product (Hoffer, 2011). By combining the two methods, we aimed to address a larger spectrum of required project activities.

Table 1 provides a complete overview of all the practices that were considered during the selection. We reviewed the agile practices for their applicability in an EIS implementation. While reviewing we came across some agile practices that were very similar to each other. For this reason, some of these

practices were combined into a single practice, which was used in the rest of the development process. Table 2 provides the names of the XP and Scrum practices that were considered similar to each other. In the first column, the name of the XP practice is given and the second column provides the similar Scrum practice. The third column provides a description of why they are considered similar. The names in **bold** indicate the names that are used for these practices for the rest of this report.

Table 2: XP and Scrum practices that are similar to each other

XP practice	Scrum practice	similarity
Small releases	**Sprint**	Both practices specify that work should be split over a number of small cycles that deliver a piece of working software.
Acceptance test	Definition of done	Both practices specify the criteria to which a user story should comply in order to be labeled as completed
Task board	Sprint backlog	Both practices specify that tasks within a sprint should be tracked on a board that tells: the status a task, who is working on a task and to which user story a task belongs.
On-site customer	**Product owner**	Both practices are about a person that determines what features a product should deliver, what the priority of those features are and have decision authority.

4.2 Refining the AIM using Semi-Structured Interviews

After this initial review, semi-structured interviews were held with four company experts in order to come up with a final set of practices that are applicable within an EIS implementation. The involved experts were selected based on their different roles and levels of experience. Two of the experts were senior consultant, one with 9 years of experience and the other with 5 years of experience. The other two experts were (i) a project leader with more than 25 years of experience, and (ii) a team leader who has about 3 years of experience in such implementation projects.

These semi-structured interviews started with explaining how the practices listed in section 2.4 were used and what their benefits are. Based on those explanations the interviewees indicated which practices they thought were applicable in an EIS implementation. Table 3 gives an overview of the practices that the experts excluded from the development of AIM. The left column provides the name of the excluded practice and the right column provides the reason of exclusion.

Table 3: list of agile practices excluded from the development of AIM.

Agile practice	Reason for exclusion
Product backlog	This practice was excluded because the 'story map' practice provided the same functionality, while at the same time helping structure the conversations with customers.
Pair programming	Although its value as an information discriminator and quality control mechanism was recognized, the associated costs of pair programming were perceived as being too high. As alternative, team members would come together on regular intervals in order to go over major decisions and evaluate each other's work.

4.3 Placing the practices within a software development life cycle

Once the final set of practices was selected; they were structured in a SDLC to indicate when to apply them within an EIS implementation project. This was done by first reviewing the SDLC of XP (Beck,

2004) and Scrum (Abrahamsson et al., 2002), because the practices originated from these methods. From this review it was concluded that both life cycles roughly had four phases: exploring and gathering requirements, planning the next sprint / iteration, executing the sprint and releasing the newly developed features. These phases, therefore, are also used in the new model.

In the next development step, the following activities were selected: “write story cards”, “create release plan”, “make sprint plan”, “requirements & analysis”, “design”, “test”, “demonstrate”, “retrospective” and “system test”. These activities together with their major outputs were placed within the previously defined phases. This general framework is shown in Figure 6. With this framework in place, it was determined which practice should be used in which activity or phase, the result of which can be seen in Figure 7.

4.4 Finalizing using a Focus Group Meeting

In order to review the applicability of the newly developed SDLC and identify any potential risk before it is applied in a real-life EIS implementation, the SDLC was presented to an expert panel in a focus group. This expert panel consisted of 5 people with different roles, representing different perspectives in the development process. One of the experts is the director of the system implementations of the Benelux, another is a project and service manager, two are principal consultants and one is a senior consultant. These experts were selected based on their significant experience (ranging from 9 to 26 years) in the domain.

The focus group was structured as follows: first, the moderator introduced the goals of the focus group in order to focus the discussion that followed. Then one-by-one, each of the SDLC phases were explained followed by a discussion. At the end of the discussion each participant was asked to list their key points of the discussion. These key points were consolidated to the issues listed in the first column of Table 4. The second column of this table holds the solutions that were formulated to overcome these issues.

Table 4: issues that were raised during the focus group

Issue	solutions
How do you educate employees, which previously only did a single step in the process, on all the skills required to participate in all of the development steps?	To solve this issue the practice of ‘pair programming’ was reintroduced. It is expected that by letting inexperienced employees pair up with more experienced colleagues, they will quickly learn the skills necessary to work individually.
How do you handle projects which have a fixed number of features they should implement for a fixed price	To handle this problem (Cohn, 2010) refers to the “iron triangle”, which has scope, schedule and resources (cost) written on each edge. This triangle is accompanied with the text: “Pick any two.” Using this method leaves only the schedule to be adjusted in case price and scope are fixed.
How do you prevent developers from simply starting configuration, before thinking of a valid design?	To ensure that developers make a design before they start configuring a specific design activity is included in the sprint phase of the SDLC.
Which people perform which set of tasks and practices?	To solve this issue the roles: manager, customer, developer and coach proposed by (Beck, 2004) were added to AIM.
How does a project that has no experience with agile adopt this new way of working?	To provide guidance on the deployment of AIM, the agile deployment framework proposed by Dingsøyr, García, Ruiz, & Piattini (2004) was incorporated into the solution framework.

Once the above-mentioned issues were addressed and adopted in the AIM, the method was considered to be applicable in practice. A complete overview of the AIM is given in Section 5.

5 The Agile Implementation Method (AIM)

AIM is a software development method that could be used for the implementation of enterprise information systems (EIS). The primary goal of AIM is to increase team's productivity and morale, the products quality and usability and customer's satisfaction. It does so by enabling incremental delivery of testable pieces of functionality into the users' hands. Providing early customer feedback, and allowing continuous discovery of system requirements. This feedback can then be used to steer the project in the right direction, increasing the likelihood of developing a system that the customer actually needs.

The method requirements discussed in subsection 2.7.1 are covered in the following parts of this chapter:

- Section 5.1 provides the set of agile practices that can be applied within an EIS implementation is given in
- Section 5.2 gives the SDLC that provides a sequence in which the agile practices should be applied.
- Subsections 5.2.1, 5.2.2, 5.2.3 and 5.2.4 cover the development aspects of an implementation.
- Subsection 5.2.5 covers the project management aspect an implementation.
- Section 5.3 provides the roles that are responsible for carrying out the practices given in section 5.1.

5.1 Agile practices

an deliver.

Table 5 provides the list of the agile practices that were included in the selection procedure discussed in sections 20and 4.2. For each practice this table provides the name in the first column, followed by a short description of how it is applied in the second column, and the third column states the benefits that the practice can deliver.

Table 5: practices applicable within an EIS implementation

Agile practice	Description	Benefits.
Sprint	AIM implements an EIS system in multiple cycles. At the end of each cycle a piece of functionality is released for user testing.	This way, problems are detected at an early stage of the project enabling a timely and adequate reaction.
User stories (US)	US capture requirements in simple terms, on (physical) cards. They indicate the: who, what and why of the feature that they capture. Good US should be: independent, negotiable, valuable, estimable, small and testable (Cohn, 2004). The size of a US should allow 6 to 10 US to be implemented within a single sprint. To get US to the appropriate size, they can be split across multiple dimensions (Lawrence, 2009).	User stories help cut the system up in pieces that can be implemented in multiple increments. It also separates the "what" from the "how" during an implementation. Due to the above mentioned properties, user stories help in determining a release planning. Finally, US promote and increase the communication with a client.

Agile practice	Description	Benefits.
Acceptance test (AT)	AT describes an unambiguous behavior of the system, which it should demonstrate given a certain context and a set of actions that are carried out on the system. ATs are written before the system is implemented and are related to a single US, but one US can have multiple AT. Generally, AT results in either a pass or fail. (Beck, 2004)	AT provides a clear goal for the developer, helping them focus on the features that the client really requires. At the same time, AT serve as an unambiguous "contract" towards the client, as it specifically states what the system should do in order to be considered complete. At also clearly indicate if a US can be considered complete at the end of a sprint, which helps determine the teams velocity.
Story points	Story points are used as a measure to indicate the size of a user story relative to one another. It completely separates the estimation of size from the estimation of duration. (Cohn, 2006)	Story points forces estimates to be made by analogy, which is shown to provide better estimates then estimates based on absolute size (Lederer & Prasad, 1992).
Planning poker	During planning poker, each estimator is given a deck of cards with values of 1, 2, 3, 5, 8, 13, 13, 20, 40, and 100 story points. For each user story that is to be estimated, a moderator reads the description and answers any questions that the estimators have about the US. Each estimator then privately selects a card representing his or her estimate. After everyone selected an estimate all estimators simultaneously shown their card. The highest and lowest estimators explain their estimates and new estimation round is held. This is repeated until consensus is reached (Cohn, 2006).	Letting estimators justify their estimates is shown to improve accuracy of the estimate, especially on items with large amounts of uncertainty (Hagafors & Brehmer, 1983). Averaging individual estimates also leads to better results, as do group discussions of estimates (Jørgensen, 2004)
Time-box	One sprint takes one to four weeks depending on the amount of user stories that need to be implemented. On the Friday of the last week a demonstration and retrospective is held. (Cohn, 2010)	By forcing a sprint to stop after a fixed period makes sure that problems in the planning are detected early. Enabling story and velocity estimates to be adjusted and the planning to be revised. (Agile Alliance, 2015)
Daily meeting	All team members stand up in a circle and answer the following questions: • What have you completed since the last meeting? • What do you plan to complete by the next meeting? • What is getting in your way? (Agile Alliance, 2015)	Daily meetings are used to promote the sharing of essential information, to pair team members that are facing the same problems, and to increase team cohesion.
Task board	A board with the columns: “to do”, in progress and done is placed in the team room. Tasks that need to be performed are placed in each of the columns, reflecting their status. (Cohn, 2006)	The task board ensures efficient diffusion of information relevant to the whole team. At the same time, it serves as a focal point for the daily meeting, keeping it focused on progress and obstacles.
Sprint retrospective	At the end of a sprint all of the stakeholders take an hour to reflect on the past sprint. Every member writes post-it’s with positive and negative events out of that sprint. Based on those post its improvement actions are listed, which should be carried out before the next retrospective (Salo, 2004).	The retrospective offers the team with the explicit opportunity to improve the team's performance over the duration of the project. It also promotes project ownership and sense of responsibility the project team has with respect to all aspects of the process.

Agile practice	Description	Benefits.
Story mapping	Story mapping is the practice of ordering user stories along two independent dimensions. The first dimension, which runs along the horizontal axis, is time. The second dimension is necessity, which runs along the vertical axis. (Patton, 2005). Horizontal lines indicate the edges of a sprint, user stories can be moved up and down these lines to indicate in which sprint they will be implemented.	The story map makes talking about user stories easier, as it logically orders them in time. The story map is also an intuitive and easy way visualizing the sprint planning (Patton, 2005).
Burndown chart	The release burndown chart shows the amount of work remaining at the start of a sprint. The vertical axis shows the number of story points remaining to be implemented and the horizontal axis shows the number of sprints that are completed. (Cohn, 2006)	Burns down charts are used as a visual indicator of how quickly a team is moving toward its goal. It helps identify problems in the planning and velocity.
Velocity	Velocity is a measure of a team's rate of progress. It is calculated by summing the number of story points of each user story that the team completed during the sprint. (Cohn, 2006)	The use of velocity corrects estimation errors, because after one sprint the number story points the team can really implement is measured. This information can then be used to adjust the planning accordingly (Cohn, 2006).
Project charter	A Project charter is four handwritten flip charts that are made within 30 minutes by all project stakeholders. The charts contain: the team values and norms, an elevator pitch for the project, a short product description, and the project success criteria. (Agile Alliance, 2015)	The project charter is used to align the effort of the team, set clear goals and to quickly introduce the project to others.
Frequent releases	The product is frequently released as a beta version into the hands of the end user. (Beck, 2004)	This is done to gain early feedback and help end users discover their needs and requirements. This also provides information on product's quality and stability.
Collective code ownership (CCO)	Every team member is indulged to make changes to "any" part of the system as they see fit. (Beck, 2004)	This stimulates a common understanding and creates a sense of shared responsibility. It should also help to increase the product quality and make sure no single person is blocking the whole project.
Coding standards	All developers use the same styles, practices and methods for configuration and documentation. (Beck, 2004)	This enables different people to work on all parts of the system and it improves the overall quality.
Continuous integration	Continuous integration aims at minimizing the duration and effort required for integration and tries to deliver a product version suitable for release at any moment. (Beck, 2004)	Due to the nature of configuring an EIS, this practice is already applied in most implementations. This is because most configuration processes are done within the general EIS and any changes are therefore immediately integrated. For this reason no additional benefits are expected.
On-site customer	There should be a customer on the same location as the development team, so he can immediately answer any questions. If the client cannot be located on site, the option of an internal product owner should be considered. (Beck, 2004)	The added value of an on-site customer offers a: • single point of information • single point of responsibility • easy and quick access

Agile practice	Description	Benefits.
Sign up	Instead of being assigned to a certain task, members of an agile team normally chose task to work on. (Beck, 2004)	Sign up increases the sense of responsibility
Common workspace	Dedicated space used by the whole team for the duration of the project, set apart from other groups' activities. (Beck, 2004)	Common team room helps with the diffusion of information between those who need it and those who have it. For instance by people overhearing others' conversations or looking at information radiators
Automated testing	Short program fragment written and maintained by the developer, which exercises some narrow part of the product's working and checks the results. There are multiple tools that can be used to provide EIS with predefined information and automatically evaluate the response. (Beck, 2004)	The use of automated testing can increase the test speed and quality, although it requires some upfront investment.
Test driven development (TDD)	TDD is a style of development that starts with writing a unit tests and then writing just enough code to pass that test. This is repeated for every feature. Once multiple features are finished and linked together, their test should be rerun. (Beck, 2004)	As with normal programming, TDD will ensure that tests for every feature get written. Additionally, writing the tests first leads to a deeper and earlier understanding of the product requirements, ensures the effectiveness of the test code, and maintains a continual focus on software quality (Pathfinder Solutions, 2012)
40-hour week	This practice states that on average a developer should work 40 hours. Overtime, other than on an exceptional basis, is seen as detrimental to productivity rather than enhancing it. (Beck, 2004)	The reasoning behind this is that overtime masks schedule problems instead of remedying the underlying causes
Refactoring	During an EIS implementation project, insights in how the system could be efficiently structured increases as a project progresses. Therefore, it is suggested to rebuild parts of the system once major improvement areas have been established. (Beck, 2004)	Refactoring improves the system configuration in terms of length, duplication, coupling and cohesion, all of which correlate with ease of maintenance. It also helps the understandability and encourages each developer to think about and understand design decisions.
Simple design	Simple design states that the emphasis should be on designing the simplest solution that is implementable at the moment. (Beck, 2004)	During EIS implementation a developer has some freedom to choose how he wants to implement a feature. Reducing this choice to the option that takes the least amount of times ensures that when changed no unnecessary time was wasted on the implementation of the initial feature.

5.2 Software development life cycle

The agile practices mentioned in Section 5.1 are placed within a software development life cycle (SDLC), in order to structure their use. A SDLC provides a set of activities conducted in a set of phases, which a project goes through when realizing a software system (Hoffer, 2011). An overview of the different phases of this SDLC is given in Figure 6. This SDLC combines the exploration and planning phases of XP (Beck, 2004) with the sprint and post-sprint phases from Scrum (Abrahamsson

et al., 2002). As with almost every agile method, the whole project is cut into smaller release cycles. Each cycle is aimed at delivering a testable set of product features into the user's hands.

The SDLC starts with the exploration phase, where most requirements are gathered in the form of user stories. This is followed by the planning phase, during which a plan is made that specifies the user stories that should be in the next sprint. This plan is executed in the implementation phase, which takes place in a time box of 1 to 4 weeks. Once this period ends, the release phase starts. During the release phase, the finished user stories are demonstrated and a retrospective is held. Finally, the implemented features are handed over to the customer so they can perform an acceptance test of their own.

Throughout the process, new requirements or bugs may be discovered. These are captured as new user stories and taken into consideration during the next planning phase. This way, the process keeps iterating over planning, sprint and release, until no new requirements are discovered or a predefined time frame has elapsed.

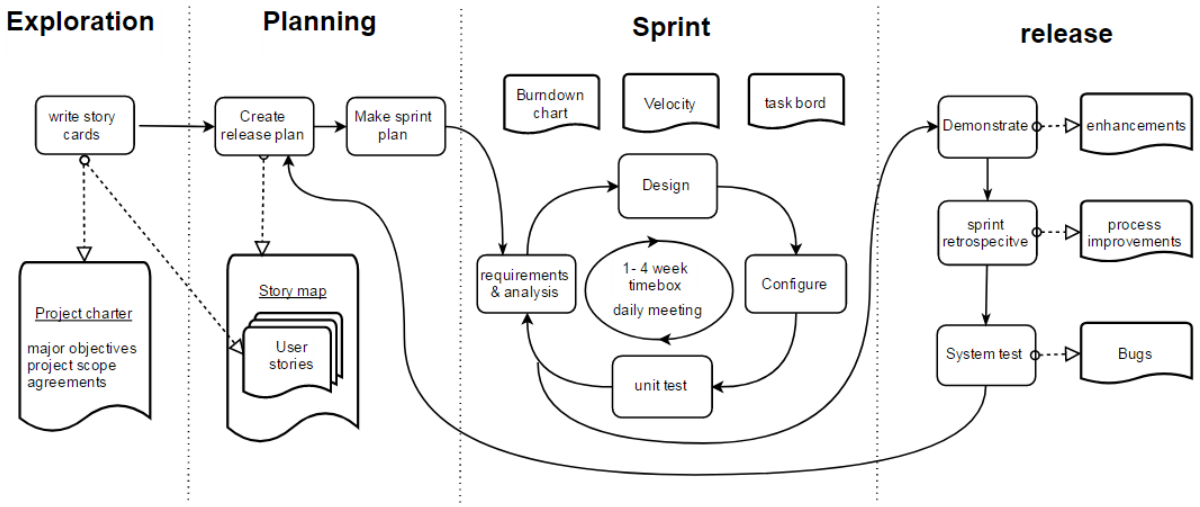


Figure 6: AIM process model

5.2.1 Exploration

During the exploration phase client sessions are held in order to get a broad feeling for the deliverables, major features, and functionality required. During the first client session, a project charter is prepared in 30 minutes on 4 flip chart pieces of paper. This project charter should contain 4 elements. First, it should state the team values and norms that need to be uphold during the project. Second, it should provide an elevator pitch that sells the product that is to be created. Third, a short description of the product, with key selling points should be provided. Finally, the success criteria of the project should be captured.

During this phase system requirements are captured in the form of user stories, acceptance tests and constraints. Constraints specify quality attributes to which the system must comply. Quality attributes could for examples say something about the performance, efficiency, reliability, robustness, safety or security of a system. Acceptance tests and constraints are always associated to a user story. User stories are placed on a story map to structure them logically relative to one another, increase their understandability.

5.2.2 Planning

During the planning phase a release planning and sprint planning are made. This is done by estimating the size of each user story in a number of story points. When known, the development team provides their velocity of previous projects. If this information is not available, an estimate is given. Using the size estimates and velocity, the amount of story points one sprint can implement is determined. This amount, together with the priority of user stories, determines which user stories will be implemented in the next sprint. In the case of an inaccurate velocity or effort estimates, the team makes a re-estimation and adjusts the set of user stories to be implemented in one sprint accordingly.

When choosing user stories for the next release, one should take their dependencies into account. The aim should be to implement those user stories that together provide end-to-end functionality to the costumer. Another aim should be, to start the first sprint with those user stories that forces the developers to build a general architecture of the product. This will then enable the subsequent sprint to incrementally build and improve on that architecture.

When the release plan for the following sprint is made, the team determines which tasks need to be performed in order to implement the user stories. The developers select a set of tasks for which they will be responsible in that sprint. Figure provides a detailed overview of the different steps, agile practices, outputs and stakeholder's one iteration contains.

5.2.3 Sprint

At the beginning of the sprint, the developer analyses the user stories, acceptance tests and constraints that are associated with the task s/he is working on. Based on these inputs, s/he designs a solution. This design stage forces the developers to think about the global structure of the system, before starting the actual configuration. The resulting design document will also offer a quick system overview that will speed up and increase the quality of any changes that need to be made later on in the process. Another advantage of this design document is that it allows work to be handed over between developers more easily. When the design has been made, the system is configured accordingly. After configuring a piece of functionality, the developer runs a unit test in order to see whether the functionality shows the required behavior. If the test fails, the developer may go back to any of the previous steps in order to fix the problem. When, after a couple of tries, all features associated with a user story are implemented, the acceptance tests of that user story can run. When all the acceptance tests are passed, the user story is set up for the upcoming release. This also means placing these user stories in an environment that lets the customer start performing the user acceptance test of all newly finished user stories. Any bugs or enhancement that are discovered are captured as new user stories, which are taken up in one of the following sprints.

5.2.4 Release

When the time-box of the sprint ends, the essential information needed for the system maintenance is documented. Based on the completed acceptance tests, the user stories that are considered to be finished determined. This information is used to determine or adjust the team's velocity. All finished user stories are then demonstrated to the customer. The customer provides feedback to the development team based on this demonstration. Any deficiencies that are detected are captured in the form of enhancement user stories. After the demonstration, all stakeholders participate in the sprint retrospective. The retrospective will follow the post implementation workshop (PIW) method proposed by (Salo, 2004) and the action point list from (Salo, 2005). During a PIW the project team

generated positive experiences (i.e. the practices that should remain the same) and negative experiences on post-it notes and placed them individually for display on a flip chart with clarifying comments. The findings are grouped and labeled, after which the emerged negative topic areas are discussed to come up with improvement suggestions. The post-iteration workshops end with the generation and agreement on the improvement actions list for the next sprint. The action point list (Salo, 2005) includes the following 5 topics for each improvement action:

1) the exact problem that the action point aims at solving,
2) the specific action to be taken,
3) the responsibilities for implementing the action and schedule,
4) the means to validate the usefulness of an action point, and
5) the results (qualitative or quantitative) of validation (updated in the following PIW after piloting)

This retrospective meeting is led by a facilitator who will take little part in the discussions, but will focus primarily on creating the conditions for an effective group discussion.

5.2.5 Project management

Throughout the project, several scrum practices are adopted for project management purposes. One of those practices is daily meetings. Daily meetings are held to keep track of progress and to identify (blocking) problems in the early stages of a sprint. The progress of the team is also visualized in the form of two burn-down charts. One burn-down chart keeps track of the progress of the sprint and the second one tracks the progress of the entire project. The later also shows the velocity that the team has. This velocity is used after the first release planning to determine how many user stories will be implemented in the next release. Next to burn-down charts, a task board is maintained to keep track of the status of the user stories that will be implemented during the sprint. This task board supports the conversations in a daily meeting. At the end of a sprint the task board is reset.

Figure 7 gives a detailed overview of; all the stages a single iteration cycle goes through, who is involved in those stages and what the output of those stages are.

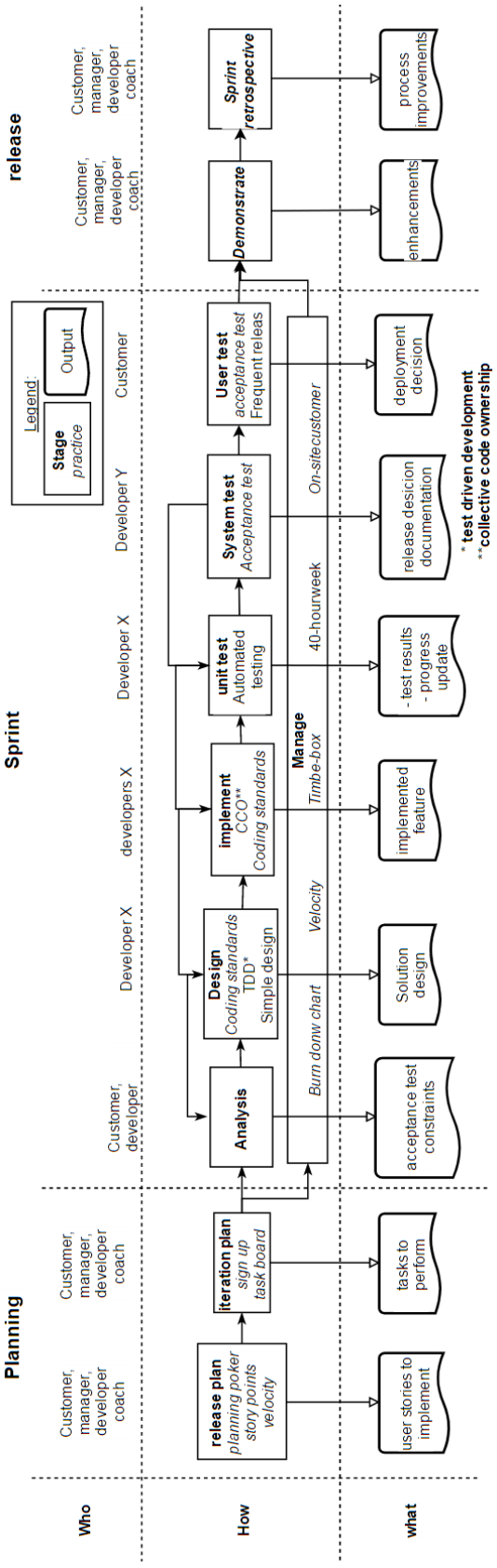


Figure 7: A single iteration cycle overview

5.3 Roles and responsibilities

This section provides descriptions of the different roles and responsibilities within an implementation team. These roles are mainly based on the existing roles in extreme programming (Beck, 2004), with adaptations based on Scrum (Cohn, 2010).

5.3.1 Developer

The developer is the person responsible for realizing the eventual system. But in order to do so s/he first needs to help the client in developing user stories and acceptance test so that it clear to him what it is that he needs to realize. During the release planning, the developer estimates the effort that each user story will take to implement. Then, based on the user stories that need to be implemented in the next sprints, the developers identify the tasks that will be needed to transform a feature request into working and tested software. Only after getting assigned a set of tasks does the developer start the actual realization of the product. After delivering a piece of functionality the developer is also responsible for documenting the essential information that the client requires for the maintenance of that functionality. Throughout the project the developer is also responsible for identifying any risks and impediments of the project and making them known to the manager. But by far the most important task of the developer is to communicate. In order to understand the client's requirements, to gather and share knowledge with team members, to discuss problems with the manager and coach, etc.

5.3.2 Agile Coach

The agile coach should have extensive knowledge on agile practices and how they are applied. At the start of a project, he should discuss with the development team about the practices that will and will not be applied. During the project, the agile coach needs to ensure that the project is carried out according to the selected agile practices and their rules. In case of problems with the development process, the coach should be able to identify the agile practices that help in that situation. In addition, he observes the team and intervene only when a significant problem is foreseen. This is to ensure that the team becomes self-reliant.

5.3.3 Manager / scrum master

At the start of the project, the manager should plan and supervise the exploration meeting. She needs to make sure that the right people with the right knowledge and skills are present and that they know how the project will be structured. During the exploration meeting she should make sure that a clear scope with concrete milestones are set and that mayor risks are identified. After the exploration meeting, the manager is responsible for scheduling and conducting the planning meetings. During the meeting, she provides feedback on the accuracy of previous estimates, such that the developers can improve them in the future. At the end of the planning meetings, the manager needs to ensure that there is a commitment towards a schedule with a specified set of features. When the developers start with the implementation, the manager traces the progress and evaluates whether goals are reachable within the given resource and time constraints. When goals should be adjusted, she is responsible for discussing this with all of the relevant stakeholders and for coming to a mutual agreement. Finally, she should remove or mitigate any identified risks and impediments, such that the team productivity is optimized.

In order to visualize the progress of the development team, the manager maintains the burndown charts and makes sure the team updates the task board. She also conducts daily meetings and tracks the teams' velocity and effort estimates, which she uses in the next release and sprint planning.

The manager role resembles that of the Scrum master in the Scrum process, except that she is not responsible for the agile practices. In small projects it might be best to have one person with both of those responsibilities.

5.3.4 Customer

The customer should be a person with extensive knowledge on the product that is to be created. Together with the developer, the customer must capture this knowledge in the form of user stories. The customer should also formulate acceptance tests, that when run successfully, ensures the systems meets their requirements. When required, the customer should also be able to explain and answer any question about the user stories and acceptance he wrote. This also means that the customer should constantly communicate with the development team and speak as the single voice representing his company. The preferred way of achieving this is to have the customer sit in the same room as the development team. When that is not an option, it should, at least, be possible to call the customer on a daily basis. During the release planning meeting, the customer needs to be present in order to provide the priorities of the user stories. The customer should also have the authority to make any decisions regarding the requirements, planning and scope of the project. This way, when the project runs into unexpected problems, the customer together with the manager can quickly come to a solution. This might mean adjusting the scope, moving user stories to a next sprint or replacing them with higher priority ones. At the end of a sprint, the customer participates in the retrospective and runs the user test. The customer is also the person at the client site that needs to make sure that the project does not go out of scope or budget.

In the situation that the client organization is not able to provide someone who can play the role of customer, the vendor should provide someone who can play this role. This person is also known as the product owner. The product owner functions as the bridge between the development team and the parties involved from the client side.

5.4 Deployment

Before starting with the improvement of a software development processes in the organization, one should have a change or a transition plan that describes how these improvements should be put in practice. According to Aken, Berends, & Bij (2012), a change plan should involve: 1) identifying relevant stakeholders, 2) defining the objectives of the change process, 3) the steps that need to be taken to achieve those objectives, and 4) communicating the change plan with the stakeholders.

5.4.1 Identifying relevant stakeholders

From the start of the change process the stakeholders should be identified. There are direct and indirect stakeholders. Direct stakeholders are the people whose daily activities are directly changes by the new way of working. Indirect stakeholders are those people whose work does not change, but that do have some kind of authority over or political interest in the process subject to change. In case of an agile implementation process, the direct stakeholders are all the people who will fulfill one of the roles described in section 5.3. Indirect stakeholders might include: management, sales department, finance department, software suppliers, HRM and any others with some sort of

influence over the project. The extent of this list heavily depends on the type of industry and project in question.

5.4.2 Defining the objectives of the change process

When making changes to a business process, the goal is not the change the process itself. Rather, it is a tool that is used to realize some business value. When deploying AIM, these business values should be close to the values stated in the agile manifesto: increasing responsiveness to change, improve customer collaboration, promote personal interaction and deliver working software that has value for its user.

5.4.3 Steps towards achieving the objectives

In order to determine which actions are needed to achieve the objectives of the change process, it is suggested to use the "agile deployment framework" proposed by Dingsøyr, García, Ruiz, & Piattini (2004). The "agile deployment framework" has 4 steps: 1) select agile practices, 2) plan deployment, 3) execute deployment and 4) analyze and package results, and improve. Each of these steps is briefly summarized in the subsection "Summary of the agile deployment method".

This method is suggested because of its fit with AIM in the following areas; (i) the framework is based around agile practices and not a specific agile method, (ii) it makes use of the iterative nature of AIM to make improvement within single project, (iii) it implements a procedure for continues process improvement on an organizational level. An organization can therefore start using a subset of agile practices that are appropriate to their specific context and adopt additional practices along the way.

However, before using this framework, other deployment frameworks should also be considers. This is to insure that the framework best suited for a specific context is selected. Section 2.6 of this report gives an overview some of the other deployment frameworks that are available in the literature.

Summary of the agile deployment method

1) Select agile practices. In the first step, the organization should set goals for deployment and then select the agile practices that fit to these goals. The agile assessment (Pikkarainen & Passoja, 2006) can be used as a systematic and goal-driven mechanisms for identifying and selecting suitable agile practices for the organization specific context. The same selection method can be applied within an EIS implementation, due to the independent nature of agile practices.

2) Plan deployment. The organization can use different approached to deploy new practices. One approach would be to experiment with a pilot project, while another might implement the new practices directly in the organization. In both cases, a plan should be made on how empirical feedback is provided and analyzed for a continuous improvement of organizational practices. The deployment phase also includes the preparation of projects involving changes to the daily software development practices.

3) Execute deployment. The Focus of the execute deployment step is to gain feedback from pilot project that is deploying the selected practices in order to enhance the organizational processes. The short development cycles of agile software development provide rapid loops, which allow project teams to iteratively improve and adapt their daily working practices based on their own experiences. This iteratively improvement is achieved by having a post iteration workshop (PIW) at the end of every sprint. The PIW enable the pilot team to tailor the deployed practices during the ongoing

project and provides the organization with a mechanism to systematically and rapidly gain feedback from the pilot project.

4) Analyze and package results, and improve. In the final step, deployed practices that have been found useful in the pilot projects are identified and employed in the organization. The organizational level can gain process knowledge from the agile assessment and the PIWs. This feedback is analyzed, improvement actions planned and implemented, and the results stored and packaged for later software process improvements.

5.4.4 Communicating the change plan

All identified stakeholders should be made aware of the nature, timing and progress of the change plan. This requires a communication plan that specifies the ways and timing of informing the various stakeholder groups. In addition, the attitudes of each stakeholder towards the project should be evaluated. Any resistance to change should be identified and a strategy to deal with that resistance should be formulated. This report does not elaborate into the types of potential resistances and ways to handle them. Interested readers are referred to chapter 9 in (Aken et al., 2012).

6 Case study

As a design science artifact, the applicability of the AIM should be evaluated in a business environment for which it was designed (Hevner, March, Park, & Ram, 2004). Therefore, we applied AIM in a medium sized company that is looking into the benefits of transitioning from a waterfall to an agile way of implementing their EIS. The objective of this case study is to evaluate the validity of the designed artifact, in other words, we want to investigate if the artifact works in real-life settings and does what it is meant to do (Gregor & Hevner, 2013).

This case study will be presented as follows: first, a description of the organization in which the case study was performed is given. Then all of the steps that were taken during the case study are discussed. Chapter 7 gives and overview and evaluation of all the results that were gather during the case study.

6.1 Case Organization

The company chosen for the case study is an EIS vendor that has its corporation headquartered in Europe and employs more than 500 employees in over 6 countries with more than 11 sites. It provides its clients with a highly configurable solution platform that supports all of their core processes. The EIS is structured in such a way that it can be deployed modularly, providing a client with the option to either deploy per module, up to the full system. Once deployed, the EIS allows quick introduction and adjustment of new company processes. The EIS vendor also offers the EIS solution in a software as a service (SaaS) model.

All implementation projects currently deploy a waterfall implementation method. At the start of such project, an implementation plan is made that specifies multiple increments the implementation will go through. Each increment will deploy one or more core processes of the client. When this implementation plan has been finished, the system architecture is made and the infrastructure required to configure the business processes is put in place. Once the infrastructure is ready, the first increment is started.

At the start of an increment the requirements of the business processes are gathered and captured in multiple documents. These documents go through a couple of improvement sprints, before the client eventually approves them. Throughout the increment this document sets the scope of what shall be implemented. Any changes to the requirements that are outside of this scope are handled according to an official procedure. Once signed, the requirement documents are translated into multiple technical designs that specify how the system should be configured. During system configuration both the initial requirement and design document are used as inputs. After completing the configuration an internal regression test is performed and any bugs that are found are solved. Once the bugs are fixed the client also performs a regression test. This test usually results in a number of additional bugs and enhancements, which need to be addressed before the EIS can go live.

When an increment is completed the next increment is started. Usually multiple increments run in parallel. Once all of the processes of the client have been implemented and the mayor bugs have been dealt with the EIS goes live. Once live, the system is maintained and SLA's are managed.

The EIS vendor currently faces multiple problems within one increment. One of the problems is that a client does not know the exact requirements at the start of an increment. This causes incorrect requirements to be documented. Another problem is that the client does not thoroughly read or understands the requirements document, due to its length and technical nature. This combined with a changing business environment and customer requirements, causes the requirement document to be outdated soon after it has been created. However, as the acceptance tests only happen after the whole process has been implemented, this discrepancy is discovered only in the final stages of an implementation increment. The bugs and enhancements that are introduced at this stage are very costly to implement, due to the high number of dependencies.

The goal of the EIS vendor is to deploy a set of agile practices in a pilot project in order to utilize those experiences in evolving an organization specific agile process model that would solve the above mentioned problems.

6.2 Application of AIM

To introduce the agile practices from AIM within the case organization we used the agile deployment framework proposed by Dingsøyr et al. (2004). This method was chosen for the reasons mentioned in subsection 5.4.3. This subsection gives a brieve discription of how each step of the agile deployment framework was executed.

6.2.1 Selecting the AIM practices

The first step in deploying AIM within the case organization was to select the practices from AIM that they wanted to start using. As the case organization participated in the development of AIM, they already had a solid understanding of what agile practices they already used and which they wanted to try. The (elaborate) agile assessment proposed by Pikkarainen and Passoja (2006), was -therefore deemed unnecessary.

The practices that the case organization was already applying were: "coding standards", "continuous integration" and "40-hour weeks". These would also be applied during the pilot project. The practices that the organization would not apply in the pilot project were: "sign up", "automated testing", "test

driven development", "planning poker" and a "common work space". The reasons that these practices were excluded are listed in the Table 6 down below.

Table 6: Excluded agile practices

Practice	Reason for exclusion
sign up	It was determined that the team members of the pilot project, did not have the experience yet with all steps of the implementation process, required to select their own tasks.
automated testing	The case organization did not have an automated testing tool yet. Acquiring and setting up a tool was deemed too expensive and time consuming for the pilot project.
test driven development	Because a large part of test driven development is based on automated tests, this practice was not applicable without the automated testing practice put in place.
planning poker	Because only one of the pilot project members had experience with estimating the size of work, it was determined that this member would make the size estimates. Another reason was the amount of time the planning poker would take.
common work space	Due to team members working in different countries, it was not possible to work in a common team room.
on- site customer	Due to the current relationship that the case organization has with the customer, and the expectation they have, it was not possible to have them on site. Instead, an agreement was made that they would be available on the phone in the period that the team would work on the project.

6.2.2 Plan deployment

Once the practices were selected, the second step was to get managerial approval for the pilot project. This was done by identifying the relevant stakeholders and involving them in the process of applying specific agile practices within the companies' context. This enabled them to list any constraints and state any concerns they had. The following constraints were listed:

- The project should be relatively small, minimizing the impact if things would go wrong.
- The product needed to be well-known in order to insure that a good level of quality would be delivered.
- The standard documentation needed to be made,
- The project should be completed within the original schedule and budget.

Once the approval was given, different project managers were approached in order to look for a client who would be willing to participate in the pilot and had a project with appropriate characteristics. The client that was eventually selected was a company established in 2001 that provides insurance services with its 165 employees. The company provides insurances products to employees and retirees in railway-related businesses, their spouse / cohabitant, children and grandchildren. This segment is appropriate for some 72.000 policy holders, which together have close to 242.000 insurances. From this point this client will be indicated as "insurer".

Before the start of the case study, this insurer was one month into the implementation of the new EIS. The generic parts of the system were therefore already put in place. What was left to be done, was the implementation of specific insurance products. The general structure of these product (e.g. the premium and coverages) were already known to the insurer. Wat remained to be done was the

UI / UX configuration and some automation. Eventually, the implementation of the boat insurance product was selected for the pilot project, because the most important parts of this type of insurance product were already standardized within the EIS. Al of the above mentioned factors took away enough uncertainties for “the company” to try out a new implementation method.

Once this selection was made, the insurer was approached to discuss the implications of the pilot and to agree on the roles and responsibilities each of the companies would have. Together with the client, a global schedule was agreed upon. This schedule specified that the project would start on 02-05-2017 with an exploration meeting and planning meeting that would together take 4 hours. After that meeting, every two weeks a new meeting would be held, during which the demonstration, retrospective and planning of the next sprint would be held. These meetings were scheduled to take 2 hours.

In between meetings, a team of 4 people each would work 8 hours per week to implement the product. These 4 persons had different levels of experience. One was a senior consultant that has worked for 9 years at the EIS vendor. Two other team members were consultants, one for 4 years and the other for 2 years. The last team member was a junior consultant with 1 year of experience. Two of these team members were living in the Netherlands, another in Norway and the last in Sweden.

The senior consultant acted for the roles of both manager and developer. The other team members performed the role of developer. From the insurer’s side, a product consultant with 16 years of experience took on the role of product owner. From this point forward, team members will be indicated by either “manager”, “product owner” or “developer”.

6.2.3 Execute deployment

The third step of the pilot project was the actual execution of the deployment. This subsection highlights the most interesting observations made during the pilot.

6.2.3.1 *Project preparation*

The first thing that was done was the creation of an initial set of user stories and acceptance tests (see appendix C for the template that was used). This was done by the manager and the author of this report before the project officially stared.

6.2.3.2 *Exploration meeting*

The exploration meeting started with setting the goals of the meeting, followed by the creation of the project charter (see Appendix D: Project charter for the results). During the construction of the project charter the focus was on how the team should work together, how the agile method would be adopted, and which benefits that would provide.

Once the project charter was finished, the initial set of user stories was discussed. The original user stories were written from the perspective of the insurer. During the discussion, it became apparent that the end-user perspective was preferred. So, it was agreed that the user stories would be re-written and that new stories would first take the end-user perspective. Only in situations where the end-user perspective could not be applied, a different perspective (e.g. insurer or intermediary) would be used. In the end, no other perspective had to be used.

It was noted by the manager that this way of working forced the insurer to start and understand their own product. He also indicated that it took him considerable effort to keep himself on the background. Because the manager was also the person with the most experience on implementation, it was tempting for him to press his ideas on the rest of the group.

6.2.3.3 *Rerelease planning*

The first planning meeting immediately followed the exploration meeting, as this saved a lot of travel time for all the participants.

From a business value point of view, it was most interesting for the insurer to start implementing the most elaborate insurance product. However, due to its size this product could not be implemented within a single sprint. Moreover, this elaborate product contained the same parts (and more) as the other two insurance products that needed to be implemented. For this reason it was agreed upon that the first sprint would start with the least elaborate of the products and would then be extended in the upcoming sprints. Appendix B gives an overview of the user stories that were selected for the first sprint. The following release planning meetings took less time than the first. The team would start with adjusting the velocity and add new user stories to the backlog. After prioritizing the new user stories, a new set of stories were selected for the next sprint.

Table 7 gives an overview of: the amount of user stories were selected for a certain sprint, the sum of the story points those user stories had, the amount of story points that were finished in that sprint and the amount of changes / bugs that were found.

Table 7: overview of planning outcome

Sprint #	Duration (weeks)	Number of Stories	Amount of story points	Story points finished	# changes / bugs
One	2	10	37	32	2
Two	2	15	37	33	2
Tree	2	19	44	24	1
Four	2	22	40	28	3

6.2.3.4 *Stand up meeting*

During the first stand up meeting the issue and project tracking tool Jira (see appendix B for screenshots) and share point were demonstrated, because no tasks could yet be discussed. During this meeting the each user story was discussed, briefly stating what should be delivered and which steps would need to be taken in order to implement them. After this explanation, developers chose which user stories they would work on that week and were given the opportunity to ask questions.

The other stand up meeting followed the regular stand up procedure as discussed in section 5.1.

6.2.3.5 *Demonstration & Retrospective*

During the demonstration the user stories that were finished were shown on screen and the customer was asked to write down any things she was missing or would like to see changed. This sometimes resulted in minor additions and minor changes to the appearance of the system. The additions were captured in new user stories and the changes together also formed one new user story. These user stories would be taken into consideration during the next planning meeting.

After the demonstration the retrospective was held. The retrospective followed the post implementation workshop (PIW) method proposed by (Salo, 2004) and used the action point list from (Salo, 2005)(see section 5.2.4 for a description of these methods). Improvement topics that were discovered during the workshops included: scheduling, information exchange, user stories, stand up meetings and reviews. (See Appendix E for the actual flip chart that was created during the first retrospective).

With the problems area's identified, the team discussed how these issues could be solved. This resulted in a set of actions that would be carried out by a specific actor and a set of plans on how the results of those actions would be validated. All these things were captured in an action point list, an example of which is shown in Appendix F: Action point list.

The second retrospective showed that some of the improvements had been carried out but others had not. The most common reason for not carrying out an improvement action was the lack of time and sense of responsibility for the action points.

As a result the improvement points were also no longer written down in an action point list. Rather they were formulated by the team as points they needed to keep in mind and they should try to follow. This had the same result as the action point list, some improvements were carried out other were not. The only difference was the amount of time that was spend on creating the improvement points, which went down.

7 Results and discussions

This chapter provides an overview of the results that were obtained in the case study. First, the results of the semi-structured interviews are summarized and discussed; this is followed by the results of the comparative agile survey; finally, the time measures are given and compared to two similar projects

7.1 Evaluation using Semi-structured interviews

The objective of the semi-structured interviews was to provide qualitative evidence on the influence of agile practices on EIS implementation project. This section starts with a brief explanation of the interview procedure and how the interview questions were generated. This is followed by a description of the interviewees and the measurement protocol that was used for extracting and consolidating the information from the interviews. Finally, the findings of the interviews are given.

7.1.1 Interview procedure

The interviews started with a short introduction, explaining the purpose of the interview, stating that the interview would be recorded and that any information would be anonymized. Following this introduction, the participant's demographic information was gathered.

Next, a set of questions about the use of agile practices were asked. Only questions about practices that were actually applied during the pilot were included. Each practice was briefly introduced by using the description provided in section 2.4. After this introduction the following questions were asked for each practice:

1. What was your experience with this practice in the project? Was this for example, positive or negative?
2. Did you face any challenges when applying this practice? If so which?

3. How do you think that this practice affected the outcome of the project?

Finally, 15 questions related to the project's performance measures were asked. The performance measures were grouped in the following categories: team, product and process related measures. One example question of a team related performance question is as follows:

1. Compared to previous project you participated in, how was the team's *productivity*?
    a. Was this for example lower or higher than the other projects you participated in?
    b. What do you think is the cause of this difference?

These questions were reviewed to ensure relevancy and completeness. This was done by a dr. O. Türetken and J. Stegeman. Dr. O. Türetken has 8 years of experience in the field of EIS implementations and (agile) software development and currently is academic member of the Information Systems Group in the School of Industrial Engineering at Eindhoven University of Technology. J. Stegeman is a senior consultant at the company and has over 8 years of experience in EIS implementations. The validation also insures that the questions are formulated in a neutral way, so that the interviewer poses minimal bias and direction to potentially influence participants' perception. The complete set of questions created for the structured interviews is given in Appendix F0.

7.1.2 Interviewees

In order to capture the perceptions of all stakeholders, people from different organizations and with different roles and responsibilities were included in the semi structured interviews, thereby enhancing the internal validity of the findings (Gibbert, Ruigrok, & Wicki, 2008). The interview sample consisted of five stakeholders that were involved in the pilot. Their characteristics are shown in Table 8.

Table 8: interviewee characteristics

	Interviewee 1	Interviewee 2	Interviewee 3	Interviewee 4	Interviewee 5
Age	32	35	45	32	25
Gender	Female	Male	Female	Female	Female
Job title	senior consultant	senior consultant	Product consultant	consultant	junior consultant
Role in project	representative insurer	manager and developer	product owner / customer	developer	developer
Years of experience	7	8	2	3	1
Number of projects	A lot	15	1	3	3
Years of agile experience	0	< 1 year	< 1 year	0	0
Number of agile projects	0	1	1	<1	0
Attitude towards agile	positive	Very positive	Very positive	neutral	positive
Change in attitude	no change	no change	no change	slightly more positive	no change

7.1.3 Measurement protocol

The qualitative nature of semi- structured interviews require a large degree of interpretation, and is therefore susceptible to subjectivity (Blumberg et al., 2011). In order to reduce this subjectivity, the interviews were recorded using the free version of Amolto call recorder. This enabled the interviewer to pay full attention to the interviewee, and use available time as efficient as possible. In addition, recording the conversation enables the transcription of the interview into text, without leaving certain claims unnoticed (Dul & Hak, 2008).

To further reduce any bias, the interviews have been transcribed (see Appendix H: interview transcription) and encoded. For encoding the transcription, the coding scheme given in Table 9 was used. The left column of this table indicates the type of question that is encoded and right column provides the codes that were used for that question. These codes are Likert scale responses taken from (Sinain, 2014). If a transcription did not provide sufficient support for assigning a code to the answer, or no answer was given the following code was used: n/a ("no answer").

Table 9: Coding scheme

Question type	**Codes**					
	1	2	3	4	5	n/a
How did you experience this practice	Very negative	negative	Neither positive nor negative	positive	Very positive	No answer
Did you face any challenges	A lot	many	Somewhat	A little	Not at all	No answer
How did it effect project outcomes	Very negative	negative	Neither positive nor negative	positive	Very positive	No answer
Compared to previous projects you participated in how was the "project outcome X"	Much Lower	Lower	About the Same	Higher	Much Higher	No answer

The results of this coding have been placed in Table 9: Coding schemeof 10.9Appendix I: interview coding per factor. These coding results, together with the passages from the transcribed interviews are aimed at providing a better understanding of which agile practices are applicable within an EIS implementation and what their affects are on the outcome of a project.

7.1.4 Findings from the semi-structured evaluation interviews

This subsection provides and overview of the answers given in the semi-structured interviews. It provides a brief overview of how project participants perceived the benefits they gained in using the agile practices as well as the challenges that they faced.

For a complete transcription of the interviews the reader is referred to Appendix H: interview transcription. This subsection is structured as follows: for each topic that was discussed in the interview a summary of the answers is given, followed by a quote from the interview to provide some additional context. For each quote it is indicated in which interview it can be found. For example [Int2- manager] indicates that the quote comes from interview 2 that was held with the manager.

Agile practices

Project charter

Most the interview participants experienced this practice positively and thought that is was a good team building experience.

> *"I think it was useful so everyone had the same start, and we were on the same page and had the same meaning of what we were going to do. And it was also a way of getting to know each other a bid better than before. A good team building experience."[int3 – product owner]*

However, there were also few challenges that were reported by the interviewees. The most common one was that the agreements and criteria that we set were not visible during the rest of the project and were therefore not really used.

> *"(…) we set those criteria and then did not look at them later. Then we just slip back in our old way of working." [int5 – developer]*

> *"(…)* somewhere in the back of peoples head this information will be there, but we are not actively doing anything with it.*"[int2 – manager]*

In the end, the interviewees indicated that the positive team building experience alone was enough to positively affect the project outcome.

Planning game

Overall the interviewees experienced this practice quite neutral. They thought it was needed for this new implementation method to work, but it required a bit of getting used to.

> *"I think it is an essential part and it can be done pretty well within a configuration. Especially if you have data available with which you can determine priorities." [int2 – manager]*

> ´It *depends if you ask about the first or the last sprint (..) in the last sprint I think it's easier, because now we have the product in test and it's much easier then to see what you miss and to prioritize. In the first planning games we missed a bid of the overview." [int3 – product owner]*

There were some challenges that were faced when applying this practice - most related to getting used to this new way of working and setting the right priorities:

> *"For a company that never works agile it's difficult to have everyone understand the same thing. So, it took some time and an internal meeting to explain. And to feel like you have the right priority (…)." [int1 – representative]*

However, because it helped focus the project on what was most important and therefore spending time more effectively, all interviewees felt that this practice impacted the project positively.

> *"I think it is very good so that we don't always need to include everything in the product, but that we can focus on the most important things (…)" [int1 – representative]*

> *"By looking at priories every sprint. Then the longer a user story is not picked up the more people realize it is not important and then people can really say: OK we will not do this, because it has never been important enough" [int2 – manager]*

Product backlog

Overall the participants liked the backlog and thought that it proved a good overview of the work that still needed to be done. This helped them prioritize the work for upcoming sprints and enabled them to gather relevant information

> *"It is good to see in each sprint there are fewer user stories. But then it helps to get a bit clearer picture a bit earlier and to see the reason why we prioritize some things and why others are in the backlog. It helps with the complete overview and complete understanding of the whole team. To see what is coming more later on and to see what we need to work on and what information we need to collect." [int1 – representative]*

The only remark from the insurer was that at the start of the pilot not all of the requirements were in the backlog. This made it harder to prioritize and unclear about parts of the product that still needed to be discussed.

> *"In the beginning there was not that much on the backlog. That made it difficult to see the whole product. If the backlog was filled from the start that would have helped" [int5 – developer]*

Sprint

Most people experienced the 2-week sprint period as a suitable amount of time in which to deliver some results. They experienced the end of the period as a sort deadline that triggered them to start working on the project. The fact that something would be finished at the end of this period was said to be motivational and made working on the project fun. It also provided the client with lot of opportunities to provide feedback and a moment to change their requirements.

> *"I think it is good both for us and* insurer *to see fast results and to see the system evolving and that you do all of the steps from doing a design to implementing it. So, it's fun to complete something and not just do 10% and then hand it over to someone else." [interviwer5 – developer]*

> *"That we can see the results after two weeks helps for the next sprint and for the development itself. It helps us to give more relevant feedback and high quality of the product." [interviewer3 – product owner]*

There was one major challenge that was stated by almost all of the interviewees. This challenge was that it was hard to find the time to work on this project. This was due to the pilot being a part time project. The pilot participant, therefore, had another project they needed to work on. This made prioritizing which project to work on very hard, resulting in most work being done in the end of the sprint.

> *"There were some difficulties with organizing work in such small periods of time. In terms of planning it was a challenge for everyone. First, we tried to make everyone work on the same moment of the week, but because people have other work as well they just worked whenever they had a moment, and some work was done very late in the sprint. Because people were then like: ow wait I need to spend some hours on this, so let's do it last minute."*

Daily meeting

The daily meeting was experienced as neither positive nor negative. The product owner felt included in the team and could quickly provide relevant information for the developers and it was a trigger to start working. However, because the pilot took only 8 hours per week the meetings were only held once per week. Therefore they did not help with solving problems or letting people collaborate.

> *"It was another trigger for people to get active. It has not helped with collaboration or discussing problems, because we did not do as many of them. (…) when there are multiple meetings the chance of someone having a problem he or she can discuss increases."*

One other challenge that was faced when applying this practice was the distributed nature of the pilot. This forced the meeting to be held via video conference, which reduced the meetings effectiveness even further.

> *"We did it with audio only and with someone sharing a screen. I think it's better if you see someone's faces. Real life would be best but video is better than no video" [int2 – manager]*

So the combination of audio conferences and low amount of meetings resulted in a low amount of information to be shared and almost no problems to be solved. With the only positive effect being the increased transparency and inclusion of the product owner, most interviewees thought that this practice had no influence on the outcome of the project.

Sprint review

This practice was experienced as one of the most positive things in the entire project. People felt motivated because they could see results so quickly. It helped with spotting bugs, coming up with requirements, setting priorities and having effective discussions. On top of that, most interviewers faced no or minor challenges when applying this practice.

> *"It was really good to motivate people. To see the result and quickly sport mistakes or improvements that need to be made. When you see it together as a whole group you can immediately discuss it and find a solutions, which is much more efficient than waiting until the costumer finds it during testing" [int2- manager]*

On average the project participant therefore thought that this practice positively affected the project outcome.

> *"It has really been a positive experience. I mean we have a product up and running already after just a few weeks. Quality, for my point of view, is much better when I can see it life and then comment on what I see and relate the things that are to come to that instead of seeing a document." [int3 – product owner]*

Task board

Overall the task board was experienced positively. It gave people and overview of the sprint and quickly showed who was working on what. It also reduced the work of the manager, because people could manage the status of their user stories themselves.

> "*It's a good way to see the overview and status and it's easy to find your own tasks and assign them to others. It gives a good overview of your own tasks and sprints and what left in the other sprints." [int3 – product owner]*

"It's positive because this gives everyone insights were we are and it's also possible for everyone to manage the status themselves. Therefore the manager has not to do all of those things himself" [int2 – manager]

The few challenges that were reported all related to getting used to the digital tool that was used to manage the pilot, but nothing related to the agile practice itself.

"New way of working with Jira. Because its digital you only look at it every few days and not daily. It would be better to get a daily reminder" [int2 – manager]

So, on average the interviewees thought that this practice had a positive effect on the outcome of the project.

Sprint retrospective

The retrospective was experienced as neither positive nor negative. Interviewees thought it helped with getting a better understanding of each other and how everyone would like to work. This resulted in some obvious yet useful improvements areas for the pilot.

"I was a bid skeptical about it. I thought the comments would be to general. In the end it does work and people came up with improvements. It really helped to get a good feel of what people think is important and how they would like to work in the project." [int3 – manager]

"Some information I could have provided without someone asking me, because I know that it was important. That was kind of obvious, but I did not think of it myself. So it helped me with being aware of others problems" [int3 –product owner]

However, the interviewees also felt that the meeting took a lot of time and could be done more efficiently. Moreover, the improvement actions that resulted from the retrospective were not actually carried out.

"We made a list together and with all kind of improvements and then eventually nothing is done with those improvements."[int4 – developer]

Despite the large amount of time that the retrospective took, all of the interviewees indicated that this practice positively affected the project, due to improved collaboration and understanding.

"I think it helps people in working together better. Because you learn how and when people like to communicate and what kind of information they need. Compared to a normal project you would hear of those things months later. Or it would go on until it blows up and then it's a big issue. Now because we continuously talk to each other we can solve it in a team and don't need project leaders to do that." *[int2 – manager]*

User stories

User stories were experienced positively. The format that was used helped the insurer to think about the value that a feature had, which helped with setting priorities. Although the format did not help the vendor.

"For me as a developer I don't mind how it is written down as long as all the information is there." [int5 – developer]

It helped the manager in splitting up the work and enabled multiple developers to work in parallel, which was especially important for this distributed project. Having complete ownership and responsibility over a set of user stories also motivated the developers.

> *"It makes it easier to split up the work between team members and especially when we work from different locations it's important that we don't depend on each other's work."[int2 – manager].*

However, many interviewees felt that they did not have the right amount of experience to write user stories in the right narrative and to make them independent from one another.

> *"I'm all in favor of user stories, but I need some additional experience to write them correctly. But it's a really smart way to make requirements because it helps you to see / argue why something is relevant to the customer or the company. So it's very good to put things in context and make a good prioritization of the product." [int3 – product owner]*

Overall, the interviewees thought that this practice had a positive effect on the outcome of the project, but also considered that there was still much to be gained.

Acceptance testing

Acceptance testing was one of the least used practices. Developers did not look at them because they did not think it would provide them useful information and testers did not use them because they did not cover the whole process.

> *'I try to test the whole process of entering the dialogue and entering the application. Whereas the acceptance tests are quite technical for "*the company" *where you just can enter a dialogue or present a premium. For me, it is more appropriate to test the end to end process." [int1 – representative]*

Overall this practice did not really influence the project outcome or as the manager puts it:

> *"We did not spend much time on it and it did not really help"* [int3 – manager]

Story points

Story points were experienced as a useful tool to prioritize user stories. It was helpful to discuss the estimates as a team in the presence of the customer. This enabled quick adjustments of the user stories, leading to more efficient solutions.

> *"You can quickly discuss the steps required to implement a story and that helps to come to a quick estimate together. And because the customer hears our discussion they also get a feeling of how much work it is and why it is so much work and why a story is more work than another story. So it helps their understanding as well."[int2 – manager]*

> *"For example the super package, which we did not have today and the company indicated that it had many points and that makes us see: do we want a complete working product or do we want new features? (…) by going through the points together we could also alter the user stories, which affected the amount of points. This helped simplify the solution / product" [int1 – representative].*

Though, sometimes it remained challenging for all of the team members to contribute to estimates. However, by being involved in the estimating process they will probably catch on quicker than they did in the past.

> *"Because of the different levels of experience in the team that determined who was saying anything. So we could not really do it as a team. (...) people will learn quicker than they did normally, because we are sizing small things instead of big things." [int2 – manager]*

As with user stories, the interviewees thought this practice had a positive influence on the project, but that it also required some additional experience.

Velocity

Velocity was experienced neither positive nor negative, which was mainly due to inexperience with the practice. For the manger it was difficult to set an initial value and most developers thought that the value that was chosen was too low, causing them to stop working.

> *"It was hard to say in the beginning were to begin with and we need more practice to understand it and to get a grip on how much we can do. (…)* at one point I think people could have done more, because when you are so focused on the velocity and people finish their user stories they will not be inclined to do more. It might help if we would have set the velocity higher than people might have put in more effort to complete those stories." *[int2 – manager]*

On the other hand, the practice did help with setting a transparent limit to the amount of user stories that could be implemented in a sprint. It also helped with discovering bugs early and highlighting that a certain amount of time was needed to fix them.

> "Its new that is so visible for us so early, but we have to get used to see it so soon and I think in the total picture it's smart that we are open and see it early. Because when we are on the same team and the goal is to make a product within a certain amount of time. Then you need to find a solution together and meaty not develop something extra. [int3 – product owner]

So there were some positive and negative influences on the project, making the overall influence somewhere in between.

Team related outcomes

Productivity

Most interviewees stated that they spend less time on the project that they should have, but in the time that they spend they were more effective. Overall, the interviewers stated that the productivity was higher than the other projects they participated in. Multiple reasons were provided to justify this result:

> *"Because the tasks are smaller, its easer to finish them and easier to focus. I think people get less distracted because it is a small thing and people want to finish it." [int2 – manager]*

> *"We had little time, so we were forced to focus on what was really important and go straight to the solution in the short amount of time that we had. We also spend less time on reviewing and documentation" [int1 – representative].*

"It normally takes a lot of time to go back and forth and down details you decided halve a year ago. So, that's something we don't have to do." [int3 – product owner].

Our start reading the user stories and you get an idea of how to do it you don't have to hand it over, so that makes it quicker [int5 – developer].

Sense of responsibility

Most interviewers indicated that the sense of responsibility was the same as in other projects. The main reasons that the participants provided was that it was hard to prioritize this when they also had different projects that required their attention. This was less of a concern for the manager and product owner, who felt more responsible for the project than the other team members. However, that was also the case with the waterfall method. One interviewer mentioned that the responsibility did increase due to the small group in which the project was done.

"Everyone needed to take responsibility and everyone needed to work together quickly. This increased the accountability. [int1 – representative]

"Personally it's higher because I really believe in this way of working and I want it to be a success. (…) But one disadvantage is that I don't have the fixed deadlines, therefore the other project that did have deadlines, those projects got priority." [int3 – product owner]

Motivation

Everyone agreed that the motivation was higher for this project than in others, mainly because you could see and show your results quickly. That resulted in a lot of enthusiasm and positive feedback, which give additional motivation. However, as the product owner noted:

"I think that it's mixed, some people are used to the old way of working and like that more, because you don't have to take that much responsibility and just do what you're asked without thinking for yourself. New things are always scary." [int3 – product owner]

Knowledge sharing

Most people did not think that knowledge sharing was increased.

Everyone only did what they already could, therefor they did not need to share knowledge that much [int4 – developer]

"I'm a bid reluctant because I think we have been doing good in the other project as well" [int3 – product owner]

However, the manager, representative and product owner thought that knowledge sharing did not increase, but that it was done more efficiently:

"Due to a better understanding of how the company worked internally. This made it clearer which information they required and why. This also caused it to provide the information quicker because we knew when they would need it." [int1 – representative]

"We now have more contact than we normally have. We are talking about more things in a more efficient way. And also small issues and speak to the whole group instead of one to one. That enables you to share knowledge quickly." [int2 – manager]

"What we have done here is to include the whole value chain of the product and I think that improved the setup." [int3 – product owner]

Collaboration

The interviews indicated that collaboration did also not change compared to other projects. Reasons that are mentioned are a lack of time and the small size of tasks, which enabled people to find solution on their own.

"Everyone is working on their own stuff, only now they are small pieces instead of big ones." [int4 – developer]

"Time has stopped us to be more collaborative." [int3 – product owner]

Product related outcomes

Quality

Both manager and product owner did not think that you could compare quality in this stage of the project.

"it's a bit early to say because we are not finished and I can just compare it to the document so it's difficult to compare that to a working product"[int3 – product owner]

Those that did provide an answer thought that the productivity was higher because of multiple reasons.

In this project we don't spend a lot of time on weird / detailed stuff, therefore it's cleaner and you don't get distracted by all those thing [int 5 – developer]

"This is better because you can quickly spot mistakes and make an adjustments [int4 – developer]

Because both manager and product owner did not provide an answer, their opinion did not contribute to the average. This made the overall opinion to be "higher than in other projects".

Usability

Four out of five interviewers thought that the product was more useable. The following reasons were provided: there was more focus on simplifying the product and the problems were spotted quicker.

"We could focus on simplifying the product adding what is most important, so it is quite usable because it is simpler. " [int1 – representative]

"We discovered the useability issues quicker that in a normal project."..."this will work better, because sometimes when you are that far in a project and you find a problem in a late stage then it's too expensive and there are other priorities and it will never get changed" [int2 – manager]

Maintainability

Due to the technical nature of the questions, only the developers were asked about the products maintainability. They indicated that this would be one of the major challenges of this way of working. One of the reason was that different design solutions were chosen that would take more effort to

make changes to in the future. On top of that, multiple design styles were mixed, because different people made different parts of the system.

> *"I think we will definitely make some technical design choices at the start that we would not have made if we made a design for the whole project from the start. So, that is an issue. (..) We have to find a way to work with this, by standardizing and knowing and only using proven solution, we should be able to work like this" [int2 – manager]*

> *I think that maintaining all the documents could be more difficult because different people work on it that has different styles. That might be confusing and affect the total overview. [int4 – developer]*

Defect / bugs

It general it was believed that there were less bugs and defects found than in other projects.

> *There are less bugs because you are working on a small piece which your own responsibility [int4 - developer]*

> *I think because of the user stories and you want to following something trough. Therefore you know what should be delivered and it's not a big package. There you don't have the total overview of all the functionality. So when you develop a user story you know it should work like this and it's connected to that part. Therefor you can easily check if it's implemented correctly [int5 – developer]*

On top of that, the bugs that were found were found earlier in the project, which allowed a solution to be found while still working on the product.

> *It's the same but we find them quicker. It should help if we can solve them when we are working on the product anyway. It could be cheaper, because then you do not need to get back into the project. There is no startup cost; it's still fresh in your memory, which is not the case when the bug is found after the project is finished. So, in that sense it should be cheaper. [Int 2 - manager]*

However, there was also other responses:

> *"The number of bugs (or bug density) was varying in the other products. So, I'm not sure if it's due to agile or due to coincidence." [int3 – product owner]*

Customer satisfaction

Most people thought that the customer satisfaction would be higher, because they are more in control of the results.

> *"I think we used our time more efficient and therefore focused on more important things and working in a way that made a more focused team." [int1 – representative]*

> *"They really feel that they have something to say and to make small mistake. To them it's really explicit which choices they make." [int2 – manager]*

Surprisingly, for the customer herself (the product owner) it did not really make a difference.

I'm ok. It's like we have invented something brilliantly new. This is what we wanted it to be. [int3 – product owner]

Process related outcomes

Understandability (3)

How understandable the process was varied per person. Some people thought it was more understandable while other thought it was less. Therefore, on average the AIM was about as understandable as the waterfall method.

"This is a bit more complicated, because you need to have an overview of all tickets and keep track of changing priorities." [int2 – manager]

"The agile way of working is almost easier to understand even with a whole year of experience in waterfall process." [int3 – product owner]

"Not really understandable because I do not know all of the names used for the agile practices, but that because I did not spend any time on them" [int4 – developer]

Flexibility

All respondents agreed that this way of working was more flexible then the waterfall method. There was an open dialogue and a feeling that everyone was on the same team, which gave the customer the felling that it was allowed to change requirements. The method was also more flexible for the developers as they were allowed to choose and prioritize their own work.

"You continuously think about requirements and you are allowed to change them, even after the third sprint. Whereas in the old situation you have to decide in four weeks and you don't have the chance to change the requirements." [int2 – manager]

"It is possible to talk to developers and discuss and see and change around. And only to know that is helpful. (…) openness and the feeling that we are doing this together and not competitors of different worlds" [int3 – product owner]

"I can select my own user stories and prioritize them myself. And I could do them in my own way. So, that gave me more flexibility. For example I could start with to boring ones and end with the fun ones." [int5 – developer]

Transparency

Out of all the project outcomes transparency was the one that, according to the interviewees, increased the most when compared to the old way of working. For the insurer, it was clear how the company worked internally and which parts of the product required much time and why.

"We knew how the company worked internally and how much time it would take and what was difficult for them in the dialogues." [int1 – representative]

"First, it was more of a black box to them and now they can be part of the process and we tell them more about technical issues and that makes it more transparent how we work and what our processes are." [int2 – manager]

This increased the team spirit and helped with solving problems together.

"In other projects, I feel like it's necessary to keep your back clean and only show a good facade and not be honest about problems and hope you can fix it. Then suddenly you have a big problem that you did not know about before. If you would have known it earlier you might have had a suggestion that might help the problem disappear" [int3 – product owner]

7.1.5 Evaluation of the semi - structured interview findings

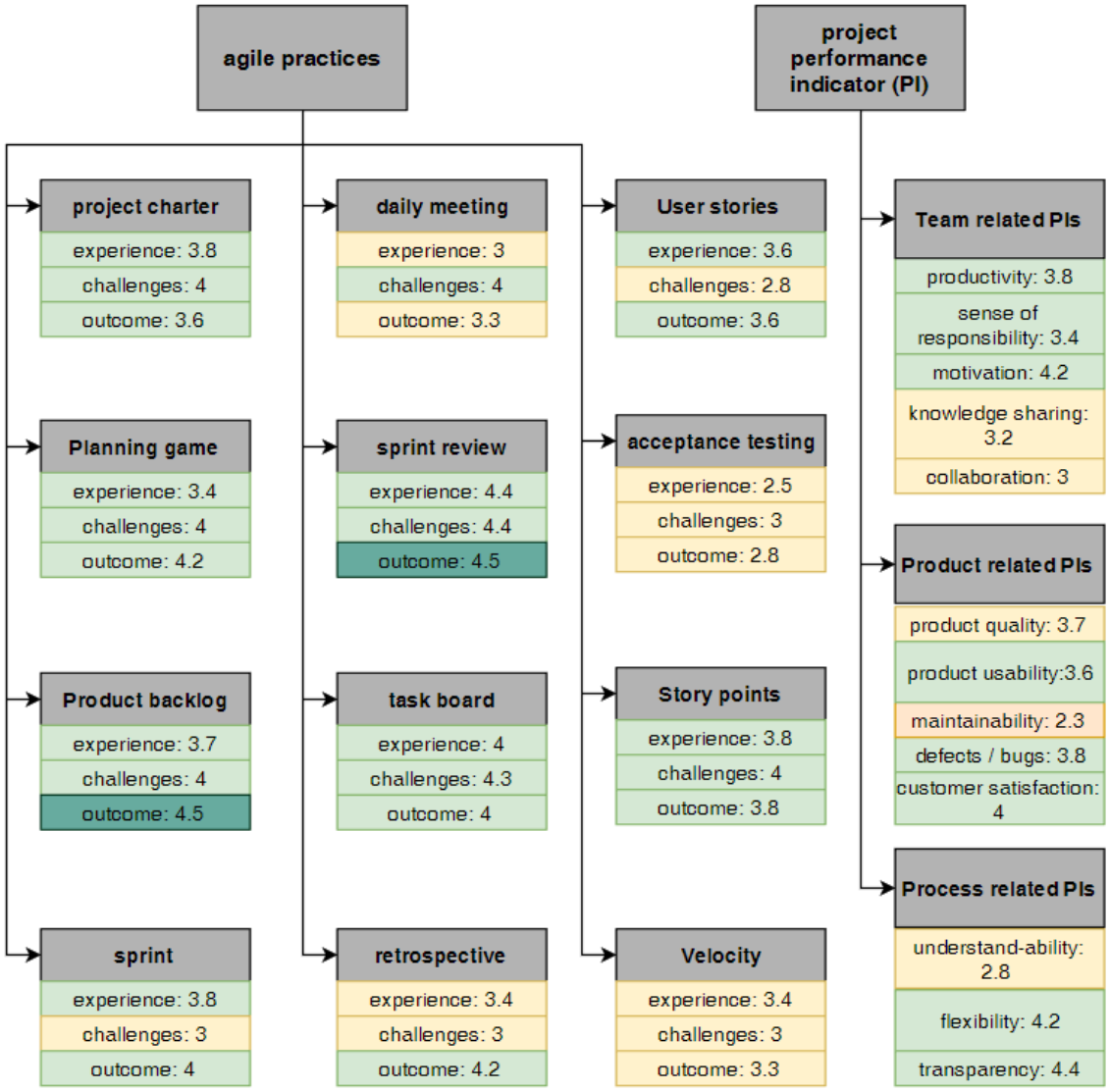
To aggregate the results of the semi-structured interviews, the codes that were assigned to the interview answers were converted to numbers using the schema shown in Table 9: Coding scheme. An interview answer coded as "very negative" would for example be converted to 1. All values for a specific question were then averaged over all the interviewees that provided an answer to that question. An overview of the resulting values is given in Figure 8: semi - structured interview results. This figure shows for each practice the average score of how it was experienced, how many challenges were faced and the effect on the project outcome. For each project performance indicator (PI), the average score of how it compares to other projects is given.

The average scores shown in Figure 8: semi - structured interview results are rounded to the nearest integer to come to a final result, which is indicated by the collor code shown in the same figure. For example the performance indicator "productivity" has an average of 3.8, which is rounded to 4. This indicates that productivity is considered to be higher then project done before the pilot.

These results indicate for each practice to which extend they are applicable within an EIS implementation and how applying these practices affect the projects performance indicators.

According to the respondents, out of all the practices the "sprint review" and the "product backlog" were the only ones that very positively influenced the project outcome. Most other practices was believed to have a positive effect on the project outcome. "Acceptance tests", "velocity" and "daily meetings" were the only practices that were considered to have neither positive nor negative effect. None of the practices that were applied in the pilot were considered to be negative or to have negatively affected the project in any way.

In total, 13 questions asked the interviewees to compare the pilots PIs with similar projects they had worked in before. On average the respondents indicated that productivity, sense of responsibility, motivation, product usability, customer satisfaction, flexibility and transparency were higher than other project they had worked in. The respondents also indicated that knowledge sharing, collaboration, product quality and process understandability were about the same as other project they had worked in. The only PI that was considered to be lower in the pilot than in other projects was the product maintainability.



Code range	5 - 4.5	4.4 - 3.5	3.4 - 2.5	2.4 - 1.5	1.4 - 1
experience / outcome	very positive	positive	neutral	negative	very negative
challenges:	not at all	a little	somewhat	many	Alot
PI	much higher	higher	about the same	lower	much lower

Figure 8: semi - structured interview results

7.2 Evaluation using the Comparative agility Survey

This section describes the results of the comparative agility survey. The objective of the survey is to assess the agility level of the pilot project and compare it to industry benchmarks. The comparative agility (CA) survey that is used is assessment tool, developed by three of the four authors of (Doyle et al., 2014) and is available on: "https://www.comparativeagility.com".

CA assesses the agility of a project on the following eight dimensions: Teamwork; Requirements; Planning; Technical Practices; Quality; Culture; Knowledge Creating; and Outcomes. The survey consists of 65 statements, each of which is related to an agile practice. The respondent must indicate the extent to which the statement is true for their team or organization. For example:

- Teams communicate the need to change release date or scope as soon as they are discovered.
- Effort spent on planning is spread approximately evenly throughout the project.

Respondents could choose an answer from the following 6 point Likert scale: not applicable, false, more false than true, neither false nor true, more true than false and true. Individual responses were excluded from analysis when 'Not Applicable' was chosen. The remaining responses are assigned ranks of 1, 2, 3, 4, and 5 respectively for all analyses.

The results of the survey are compared to the benchmark that is available on the above mentioned website, which is based upon 2,229 completed CA surveys taken between March 26, 2011 and October 12, 2012. A broad range of industries participated in the CA assessment. Thirty-one (31) different industries were identified by respondents, with the top five reported industries being:

1. Web/Software development (968 respondents),
2. Manufacturing (159 respondents),
3. Finance/Banking/Accounting (132 respondents),
4. Telecommunications/Networks and (108 respondents)
5. Non - Profit/Trade Association (95 respondents).

All of the EIS vendor employees that participated in the pilot were asked to fill out the survey via the email shown in Appendix J. Four out of 4 employees responded to this email. The results were compared to the benchmark by using the analytics available on the comparative analysis website.

The results shown below are expressed in the degree to which they differ from the benchmark. A score that differs by a positive number indicates that the answers that were provided are "better than" the average answer in the benchmark. For a negative number, it is the other way around, so the answers for that dimension are "worse than" the average answers of the benchmark. Results with a divergence of 1 or higher are considered to be remarkable and a possible explanation is provided.

The results are grouped according to eight dimensions that are assessed by the survey: Teamwork; Requirements; Planning; Technical Practices; Quality; Culture; Knowledge Creating; and Outcomes respectively.

7.2.1 The teamwork

The teamwork dimension is about empowered people working together in an effective and collaborative manner. This dimension gauges a number of important elements of high performing teams such as how they are composed, internal patterns of communication and how work is conveyed from goals to concrete deliverables.

The results of this dimension when compared to those of the benchmark are given in Figure 9. The full discretion of the 14 statements, shown on the X-axis of Figure 9, are listed below the figure.

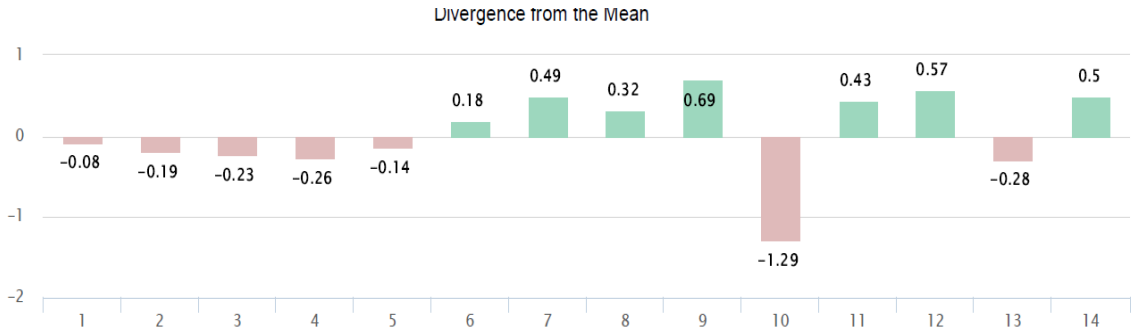


Figure 9: teamwork survey results compared to the benchmark.

List of statements:

1. Everyone required to go from requirements to finished system is on the team.
2. Specialists are willing to work outside their specialties to achieve team goals.
3. Whole teams, including the Scrum Master and Product Owner, have no more than 12 people on them.
4. People are not on more than two teams.
5. Team members are kept together as long as possible.
6. Team members choose which tasks to work on.
7. Management sets goals but doesn't tell team members how to achieve them.
8. Team members don't have to work on tasks that they deem to not add value.
9. Management rarely changes the team's priorities during iteration.
10. Team members communicate in a high bandwidth manner without undue interference.
11. Team members from one team communicate with team members from other teams in a high bandwidth Manner without undue interference.
12. Formal written documents are used to supplement rather than replace faster, more informal communication.
13. Standup meetings are effective at synchronizing work.
14. The team is not concerned about knowledge gaps when someone goes on vacation (or is otherwise unavailable).

Overall, the scores on this dimension are close to the benchmark average, with statement 10 being remarkably worse than the benchmark. The low score on statement 10 might be explained by the distributed nature of the pilot project, which caused a lot of communication to be done via audio conference.

7.2.2 Requirements

The requirements dimension addresses core concepts such as just in time (JIT) planning, emergent design and focus on customer value. Organizations that perform well in this dimension typically do not spend excessive amounts of time on activities such as documentation, meetings and extensive design sessions and instead focus on “just enough” documentation and design in order to quickly deliver value while being prepared to pivot when necessary.

The results of this dimension when compared to those of the benchmark are given in Figure 10. The full discretion of the 7 statements, shown on the X-axis of Figure 9, are listed below the figure.

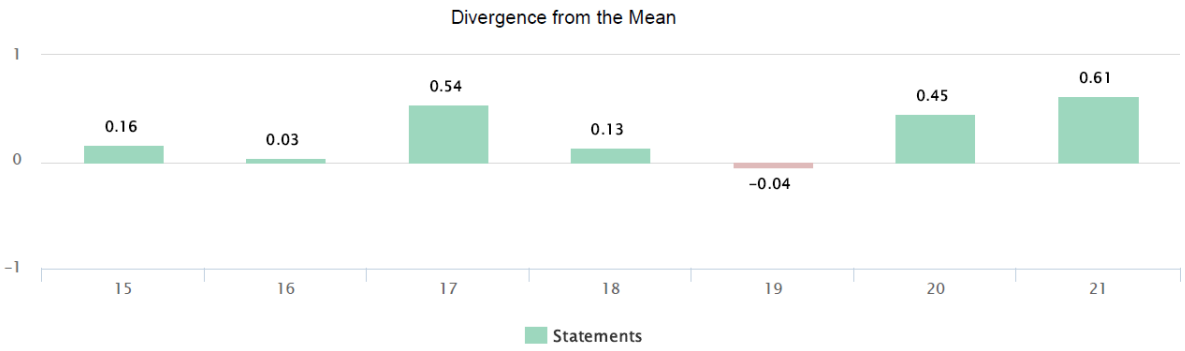


Figure 10 requirements survey results compared to the benchmark.

List of statements:

15. Teams are able to start projects with incomplete requirements.
16. Nonfunctional requirements are determined early enough to appropriately influence design and testing.
17. Requirements are represented at different levels of detail based on how soon the team expects to implement them.
18. The product owner is available to discuss upcoming features and working progress.
19. The whole team embraces change and emergent opportunities in an efficient, low ceremony way.
20. Projects do not begin with an extensive technical design phase.
21. The team performs iterative technical design throughout a project.

All of the statements were - to some extent- true in the pilot project, which is probably the reason why the mean score of the survey is close to the benchmark.

7.2.3 Planning

Having a crisp understanding of the "what" and the "why" is a critical element of business agility. This dimension focuses on fundamental success factors such as prioritization, definition of work, Product Owner involvement and establishing predictable delivery of value.

The results of this dimension when compared to those of the benchmark are given in Figure 11. The full discretion of the 15 statements, shown on the X-axis of Figure 9, are listed below the figure.

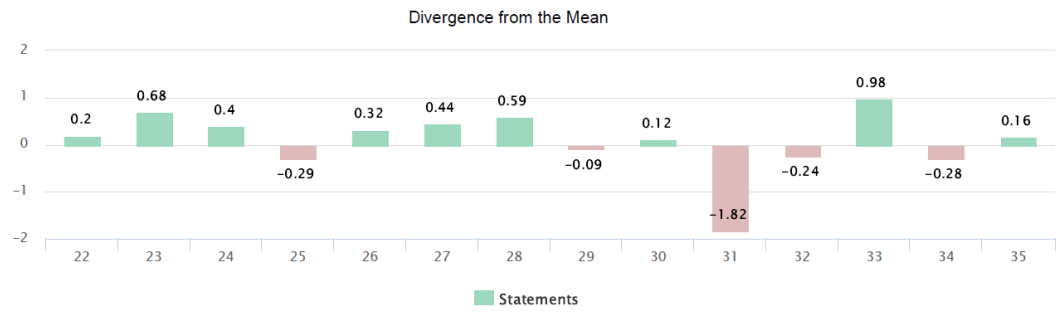


Figure 11 planning survey results compared to the benchmark.

List of statements:

22. Technical team members and the product owner are included in the planning process in a way that they can meaningfully and appropriately affect scope and deadlines.
23. Technical team members and product owners collaborate in determining what features will be included in the release plan.
24. At the start of each iteration, the team performs sufficient Just-in-time planning to be confident of what it can complete in the iteration.
25. The product owner maintains a prioritized product backlog.
26. One or more of scope, schedule, or resources is allowed to change during a project.
27. Iterations focus on creating features with value to customers and infrequently focus on infrastructure specific work.
28. All work is done in iterations of no more than 30 days.
29. Teams know their velocity.
30. There is a highly visible representation of the team's progress within a release.
31. Each day, there is a highly visible representation of the team's progress within an iteration.
32. Each feature has well-defined completion criteria that can be used to determine if the feature is done or not done.
33. We do not consider a partially completed feature done.
34. Estimates are created collaboratively by the people who will do the work.
35. Upfront planning is helpful without being excessive.

Statements 31 and 33 have the most remarkable scores of this dimension. With statement 31 having a negative divergence of 1.82 and statement 33 and positive divergence of 0.98. The negative divergence of statement 31 is might be because the pilot was not a fulltime project. On top of that, progress was tracked digitally; project participants therefore did not have to look at the task board on a daily basis.

The positive divergence could be explained in multiple ways. One explanation could be that because the project used AT, only those features that had their AT completed were considered complete. However, the answers in the structured interview revealed that the AT were barely used, making this explanation unlikely. Another explanation could be that because participants witnessed a situation in were not all features of a user stories were implemented and therefore not considered done.

7.2.4 Technical Practices

This dimension captures the degree to which software is developed in a manner that embraces emergent design, addresses technical debt and engineering practices typically associated with the principles of XP.

The results of this dimension when compared to those of the benchmark are given in Figure 12. The full discretion of the 8 statements, shown on the X-axis of Figure 12, are listed below the figure.

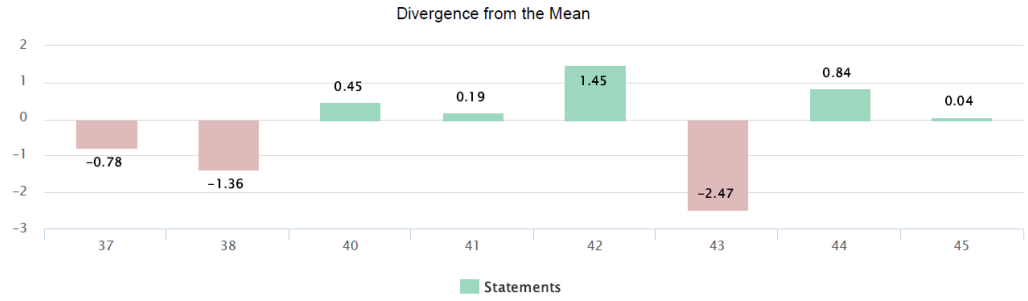


Figure 12 survey results compared to the benchmark.

List of statements:

37. Most code is written using test driven development (TDD).
38. Code is written using pair programming.
40. Refactoring is performed whenever needed.
41. Technical debt (i.e., accumulated undone or poorly done work) is made visible to both technical team members and stakeholders.
42. The entire system is built automatically at least once per day.
43. Automated unit and acceptance tests are run as part of each automated build.
44. Within our team, anyone can change anyone else's code.
45. The team can change any code in the system, even code written by other teams.

Some noteworthy differences are those of statements 38, 42 and 43. The negative value of statement 38 is probably because the pair programming practice was not applied in the pilot project. The same is true for statement 43; the pilot did not apply automated testing. Statement 42 probably diverges to such a great extent because the EIS system is built continuously during configuration.

7.2.5 Quality

The quality dimension evaluates the degree to which *quality* is built in at the source. Clearly defined customer acceptance criteria, an end-to-end testing strategy, automation and a commitment to only delivering fully tested code are all best practices that typically lead to higher software quality, fewer defects in production and less rework.

The results of this dimension when compared to those of the benchmark are given in Figure 13. The full discretion of the 6 statements, shown on the X-axis of Figure 13, are listed below the figure.

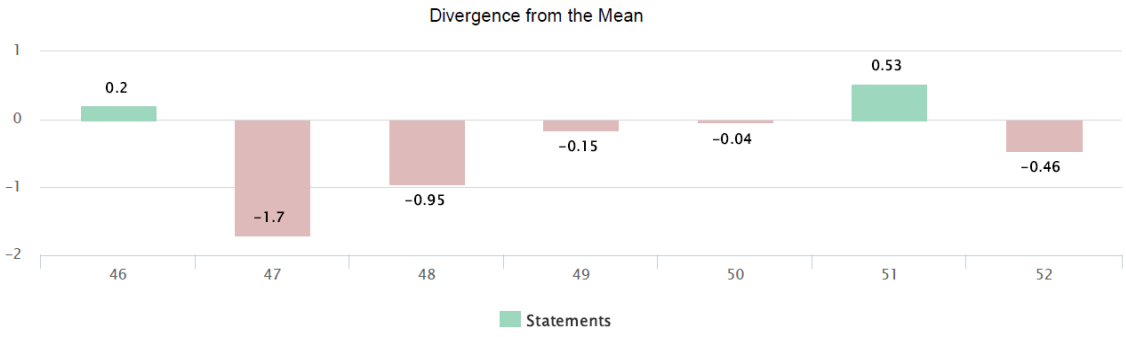


Figure 13: quality dimension survey results compared to the benchmark.

List of statements:

46. Product owners actively participate in the creation of the acceptance criteria for each feature.
47. All bugs are fixed during the iteration in which they are found.
48. At the end of each iteration there is little or no manual testing required.
49. The team performs a variety of types of testing including functional, performance, integration, and scalability each iteration.
50. Testers are involved and productive right from the start of each iteration.
51. At the end of each iteration, the team has high quality working software that it is comfortable being tested by people outside of the team.
52. The team has predefined and agreed upon criteria for considering a feature done.

There are two notable statements in this dimension: 47 and 48. Statement 47 diverges -1.7 from the mean and statement 48 diverges -0.98. The low score of statement 47 can be due to the fact that bugs were mainly fixed in the sprint after they were found. The score of statement 48 is probably related to the absence of the automated testing practice.

7.2.6 Culture

Organizational culture has an outsized impact on all other aspects of work and is a leading indicator of future company performance. This dimension covers areas such as work/life balance, the perceived degree of stress in the workplace, whether work is performed at a sustainable pace and to what degree individual compensation structures have an effect on team collaboration.

The results of this dimension when compared to those of the benchmark are given in Figure 14. The full discretion of the 5 statements, shown on the X-axis of Figure 14, are listed below the figure.

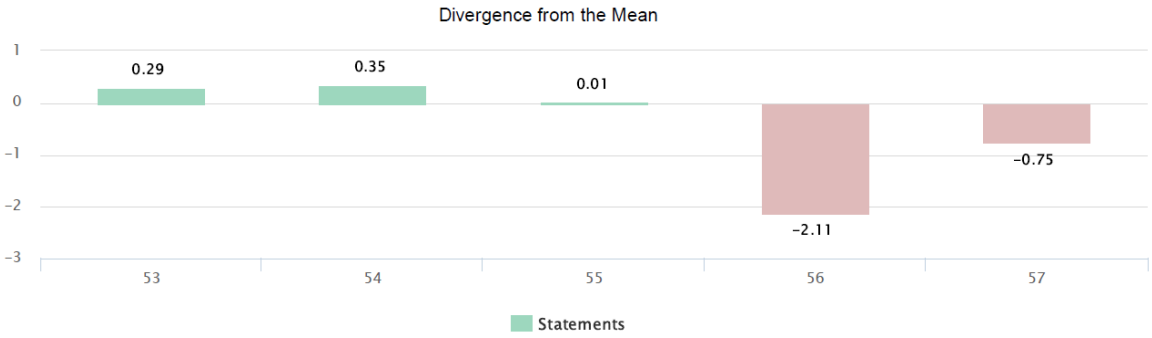


Figure 14: Culture dimension survey results compared to the benchmark.

List of statements:

53. When faced with a situation where scope cannot be met with the allotted resources in the allotted time, the initial reaction is to prioritize and explore tradeoffs.
54. The team maintains a steady rate of productivity without being overworked.
55. The team considers the economics of its choices when we make decisions.
56. Bonuses, annual reviews, and compensation promote team behavior.
57. Titles are not significant in how team members interact with one another.

Only statement 56 has a remarkable divergence in this dimension, which might be because the company's culture does not include bonuses and compensations for individual behavior.

7.2.7 Knowledge Creating

The degree to which teams are embracing the concept of continuous improvement – effectively learning from empirical evidence and their experience, evaluating existing processes and always improving the way they work.

The results of this dimension when compared to those of the benchmark are given in Figure 15. The full discretion of the 3 statements, shown on the X-axis of Figure 15, are listed below the figure.

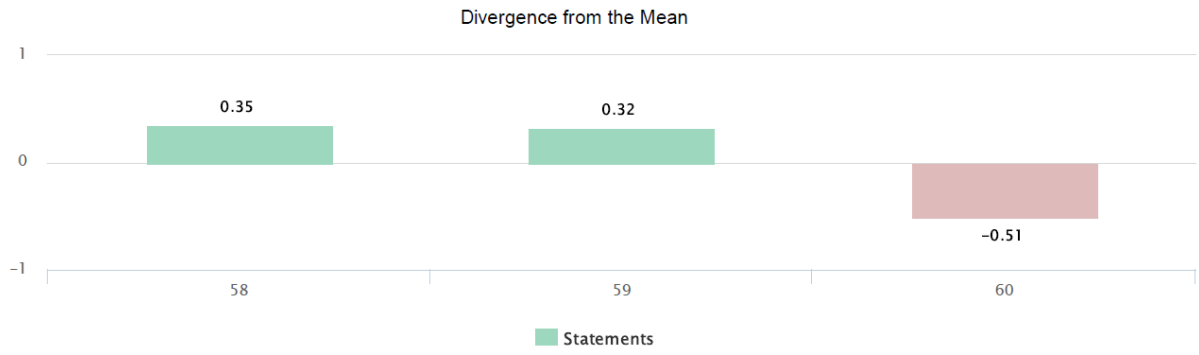


Figure 15: knowledge creating survey results compared to the benchmark.

List of statements:

58. Iteration reviews are attended by product owners, stakeholders, and team members who provide actionable feedback.
59. The team holds retrospective meetings at the end of each iteration in which the team evaluates how they are doing and discuss how to get better.
60. The team acts on retrospective feedback in a timely manner.

All statements in this dimension are around the benchmark average. Statement 60 is probably negative because the improvements that were identified during the retrospective were not all carried out.

7.2.8 Outcomes

Business agility is about early delivery of business value, adapting to changing market conditions and continuously improving the way the organization operates. This dimension captures the extent to which participants perceive Agile is making a difference relative to the way they used to work or from the last time they took the survey.

The results of this dimension when compared to those of the benchmark are given in Figure 16. The full discretion of the 7 statements, shown on the X-axis of Figure 16, are listed below the figure.

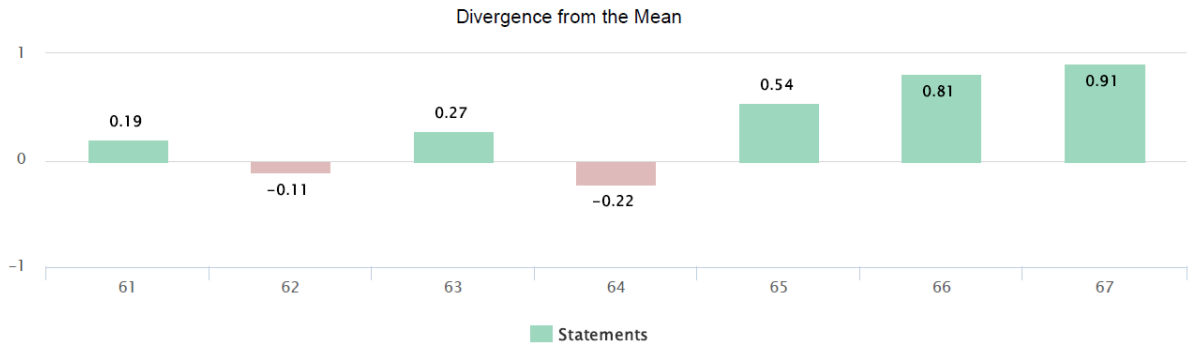


Figure 16: outcome dimension survey results compared to the benchmark.

List of statements:

61. The team is producing higher quality products than before.
62. The team is more productive than before.
63. Our customer(s) are more satisfied with the functionality of our products than before.
64. Our customer(s) are more satisfied with the usability of our products than before.
65. The team has higher morale than before.
66. Our business is recognizing greater economic value than before.
67. We are delivering functionality to users more quickly and/or more often than before.

When comparing these responses with those given in the semi structured interviews on project performance indicators, one similarities can be found. In both cases participants indicated that the outcomes improved when compared to before. This increases the reliability of both evaluations, as the two different techniques have a similar outcome.

7.2.9 Summary

To summarize the result of the survey, the average score per dimension is computed. This is done by summing the all the scores of a given dimension and dividing it by the number of statements. This results in the following averages:

- 0.02 higher on the team work dimension,
- 0.28 higher on requirements dimension,
- 0.09 higher on the planning dimension,
- 0.14 lower on the technical practices dimension,
- 0.41 lower on the quality dimension,
- 0.22 lower on the culture dimension,
- 0.5 higher on the knowledge creating dimension and
- 0.31 higher on the outcomes dimension.

The lower scores on the technical practices and quality dimension are due to certain practices that were not applied in the agile pilot. Those practices would be “refactoring” and “automated testing”. It might be worthwhile for future research to evaluate these practices as well. The low score on the cultural dimension is due to culture difference regarding bonuses and compensations.

Based on the results of the survey the following points might also still be improved; communicating without undue interfaces, holding more (daily) meetings in which progress is discussed, fix bugs within the sprint and compensate the whole team based on their results.

7.3 Evaluation using quantitative time measures

In this section, the time that was spend in the pilot project is compared with two similar projects. Those projects followed the traditional "waterfall implementation method" and also implemented a boat product. For the sake of confidentiality these implementation projects will be indicated by project X and project Y.

This section will start with a comparison of the projects productivity and finish with a comparision of the time spend per activity.

7.3.1 Productivity

In this subsection the productivity (amount of features implemented per hour) of the pilot project is compared with that of project X and Y. To do so, the requirements documents of all tree project were reviewed on the amount of: questions, information text, lists, coverages, clauses, acceptance rules, mutations, verification, calculation, output, objects and flexible field that were implemented. These attributes values are shown per project in Table 10. It should be noted that when the attribute values of the agile project were extracted, only 28 out of the 40 day project had passed (70% of the available time). This makes comparing the projects in an absolute manner unfair.

Table 10: project comparison (* these amounts are an extrapolation of the values measured at 70% project completion)

	attribute	Project X	Project Y	Agile pilot at 70%	Agile pilot at 100%*
Features	*# questions*	83	85	49	70
	# Info text	4	11	9	13
	# lists	18	14	9	13
	# coverages	19	5	5	7
	# clauses		4	0	0
	# acceptance rules	10	2	6	9
	# mutations	19	5	0	0
	# verifications	14	13	4	6
	# calculations	13	2	2	3
	# output	3	3	0	0
	# objects	4	3	3	4
	# flexible fields	1	12	0	0
Time spend	*# hours planned*	416.5	291.75	280	280
	# hours spend	396.75	291.62	145.5	208
Outcomes	*Difference in time*	19.75	0.13	n/a	72
	# features	188	159	87	125
	Productivity (# features) / hours spend	0.47	0.55	0.60	0.60

One way to in which this comparison is possible, is to use linear extrapolation (Harder, 2005) to infer the attribute values of the completed pilot project. This can be done under the assumption that time spend in the last 30% of the project has the same results as time spend in the first 70% of the project.

This wild results in the values shown in the last column of Table 10. If these numbers are correct, the pilot project will spend less time then origionally planned (72 hours) and will have implemented almost the same amount of features (125). This makes the productivity of the pilot project 0.60 features per hour, which is higher than the 0.47 features per hour of project X and higher than the 0.55 features per hour of project Y.

7.3.2 Time spend per activity

Table 11 shows the amount of time that was spend on each activities: functional design, technical design, product configuration, output and system test. But because the project cannot be compared in an absolute manner, Table 11 also shows the percentage of time spend, relative to the total amount of time spend.

Table 11: time spend on activities per project

		Functional design	Technical design	Product configuration	Output	System test	Total
Agile pilot	Hours spend	100.75	12	27	1.5	4	145.25
	Percentage	69,36%	8,26%	18,59%	1,03%	2,75%	100%
waterfall project X	Hours spend	125,44	64,25	23,25	51,75	26,94	291,63
	Percentage	43,01%	22,03%	7,97%	17,76%	9,24%	100%
waterfall project Y	Hours spend	146,75	66,00	91,00	21,25	71,75	396,75
	Percentage	37,87%	17,03%	23,48%	5,48%	18,52%	100%

These results show that the relative amount of time that was spent in the pilot on the functional design is higher than in the waterfall projects. With a smaller fraction of the time spent on technical design, output and system test then in the other projects. The low amount of time spend on output was not caused by a difference in implementation method, but coincidentally turned out to be very simple.

The relative amount of time spent on product configuration falls in between the range of the two other projects. Additional project with similar characteristics will be required in order to make a valid statement about these differences.

The relative low amount of time spend on testing could be caused by the fact that developers had to test the user stories that they had implemented. Therefore they already knew the required behaviour and could test very efficiently. Another possible explanation was given in one of the interviews:

> "(…) because they are many small pieces and you are only working on your piece, you are able to test that single piece better then all of them together." [int5 – developer]

8 Conclusion

In this study, a new EIS implementation method was developed in collaboration with expert's in the field of EIS implementations and agile development. In a number of interviews, these experts selected a set of agile practices from the Scrum and XP development methods, which were then placed within an SDLC. This SDLC was reviewed and improved during a focus group meeting with six industry experts. This resulting implementation method was applied in a case study and semi –

structured interviews, a survey and time measurements were used to answer the following research questions:

- Which agile practices are applicable in an EIS implementation?
- What are the effects of applying these agile practices in EIS implementations?

Based on the results of the semi structured interviews it can be concluded that the “project chart”, “planning game”, “product backlog”, “sprint”, “sprint review”, “task board”, “retrospective”, “user story”, “story points”, “velocity”, “acceptance test” and “daily meeting” practices can be applied within a EIS implementation. The results also showed that applying these practices resulted in a perception of increased productivity, sense of responsibility, motivation, usability, customer satisfaction, flexibility, transparency and a decrease in the number of bugs found.

The results of the survey showed that the pilot projects level of agility was close to the world-wide benchmark. Based on this, it can be concluded that the agile way of working was successfully adapted within an EIS implementation project. However, the survey results also showed that the level of agility could be improved by applying additional practices and by strictly following the rules of those practices that were already applied.

It is difficult to draw any hard conclusions on the quantitative time measurements gathered during the pilot project. This is due to the fact that the pilot project was still running when this reports was written and because there were only two similar project with which the pilot could be compared. What can be concluded is that the pilot project had a higher productivity than the two waterfall project.

Under the assumption that time spend in the last 30% of the project has the same results as time spend in the first 70% of the project, it can also be concluded that the project will complete earlier than originally planned. Finally, it can be concluded that a larger percentage of time is being spent on the functional design and less on technical design and testing.

Recommendation to “the company” based on these conclusions are given in Appendix K: recommendation to the company.

8.1 Limitations & future research

This subsection discusses the most important limitations of this research. Although this is not an exhaustive list, these limitations pose risks to the validity of this research to a certain extent. The subsequent subsections discuss these limitations, and suggest future research directions to address them.

8.1.1 Model development

The agile implementation model was developed for an EIS vendor that only operated in a particular business domain. As different agile practices might be applicable in different domains, this might affect the generalizability of the method. Future research should apply AIM within different EIS branches to validate its general applicability.

Another limitation of the model development is that some one of the experts that helped in the development of AIM also participated in the case study. This might cause this person to have some sort of bias towards the use of the model, which might have affected the findings of this study. In

order to prevent this bias, future research might consider using AIM with a team that was not involved in the development of the method.

The last limitation of the model development is the small sample of experts that was involved in the creation of the model. The six experts that were involved in the development do not accurately represent the whole population of EIS implementers. This might affect the external validity of AIM. Future research can validate the method by conducting focus group meetings with a wider range and variety of practitioners and academics.

8.1.2 Case study

The case study was performed in a single European organization that also participated in the construction of the implementation method. In addition, the case study implemented only a single product at a client who already had the general parts of the EIS up and running. Moreover, at the start of the pilot, the project scope was already clearly set and not all of the practices proposed in AIM were used. All of these issues negatively impact the generalizability of the study, since the implementation results might heavily depend on the context and the way in which it is applied. Furthermore, cultural differences with e.g. American or Asian companies might have an impact on the generalizability of the results.

Future research should consider performing research in multiple companies located in different countries and operating in different industries. They could also assess whether the method also works in larger projects that implement complete EIS from scratch. Such projects should also consider applying more of the practices that were proposed in AIM in order to see weather additional effects can be observed.

Other limitations originate from the way in which the interviews were conducted. The interview sample size of 5 and the survey sample size of 4 can be considered to be relatively low. Preferably a sample should consist of at least 30 participants (Field, 2009), smaller sample sizes effect the generalizability of the results. On top of that, the sample was not chosen randomly, which also poses threats to the external validity of the results. Then, there are the interview questions and coding scheme that were used for the interviews. These were newly created for this case study and have not been tested previously; they might therefore introduce a bias and affect the study's reproducibility. Another source of bias is that the coding process was performed by a single person, which might introduce subjectivity. In order to prevent these limitations, future research should consider performing a study in a larger project so that they can use a validated evaluation tools, like the comparative agile survey, on a larger sample size. Future research might also re-code the interview transcriptions of this research in order to validate the objectivity and reproducibility of the results.

Another limitation of this case study is that the implementation project has not yet finished at the writing of this report. In the short time frame (6 weeks) during which the project was monitored, not all potential influences can become visible and some of those would need longer time periods to manifest themselves. In addition, as the team members become more experienced with the implementation method, its effects can become more visible. All of these factors affect the external validity of the study. A longitudinal study would be able to capture the advantages that unfold only after the method is fully accepted and trusted upon.

The exploration, planning, demonstration and retrospective meetings and in setting up the workflow management tool Jira. Thus, the researcher actively contributed to changes that this research assesses and might therefore have directly created the situation for some of the conclusions. In order to eliminate the limitations of having an action-researcher, future studies should consider having a

design where the study is performed by a researcher who does not take part in the implementation project. In this way, the researcher can act more independent and objectively in the evaluation of the implementation.

8.1.3 Quantitative measures

The way in which the quantitative measures were gathered is yet another limitation of this study. Participants were only forced to report the amount of time they spent on each project activity once in every month. There were no standards as to how employees should track the amount of time they spend per activity. Therefore, it is not possible to state how accurate the time measures were and if they vary across projects.

Another limitation of the quantitative time measures is the extent to which the projects are similar. Although the same products were implemented, almost all other parameters were different. Those differences might have an impact on the time measures used in the comparison. It is therefore not possible to attribute all of the difference in the time measures to the type of implementation that is used.

In order to prevent these limitations future research should consider studying multiple projects within a similar context and use predefined way of capturing qualitative measures.

8.2 Research contributions and implications

This research provides a new EIS implementation method and qualitative and quantitative evidence on the influence this method has on an implementation project. That evidence showed that the: “project chart”, “planning game”, “product backlog”, “sprint”, “sprint review”, “task board”, “retrospective”, “user story”, “story points”, “velocity”, “acceptance test” and “daily meeting” practice could be applied within a EIS implementation. The first nine practices even had a positive effect on the project outcome. The results also showed that applying these practices resulted in participants’ perception of increased productivity, sense of responsibility, motivation, usability, customer satisfaction, flexibility, transparency and a decrease in bugs.

In comparison to most EIS implementation methods published in literature, the AIM combines agile practices originating from both Scrum and XP. This study does not only propose these practices, but also applied in a case study. The main theoretical significance of this study is, therefore, extending the findings on which agile practices are applicable within EIS implementation and what potential effects they have on a project. The results can serve as input to compare between existing EIS implementation methods and might enable academics to extend existing implementation methods or to develop new ones themselves.

From a practical point of view, this study provides practitioners with an appropriate implementation method for their EIS implementation project. This study might also guide EIS vendors that are making a transition towards the agile way of working. Providing them with a method to select agile practices from a set of practices that has been proven to work in practice and a structured and systematic way evaluating their effects.

9 References

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development methods

review and analysis. *VTT Publications*, (478), 3–107. http://doi.org/10.1076/csed.12.3.167.8613

Agile Alliance. (2015). Agile Glossary. Retrieved February 23, 2017, from https://www.agilealliance.org/agile101/agile-glossary/

Aken, J. E. van, Berends, H., & Bij, H. Van Der. (2012). *Problem solving in organizations* (2nd ed.). Cambridge: Cambridge University Press - M.U.A.

Alleman, G. B. (2002). Agile Project Management Methods for ERP: How to Apply Agile Processes to Complex COTS Projects and Live to Tell about It. *Extreme Programming and Agile Methods: XP/Agile Universe 2002: Proceedings of the Second XP Universe and First Agile Universe Conference*, *LNCS 2418*, 70–88. http://doi.org/10.1007/3-540-45672-4_8

Bebbington, S. (2014). "What is programming."

Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J. and Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved February 16, 2017, from http://agilemanifesto.org/

Beck, K. (2004). *Extreme Programming Explained: Embrace Change*. Addison-Wesley.

Blumberg, B., Cooper, D. R., & Schindler, P. S. (2011). *Business research methods* (3rd ed.). London: McGraw-Hill Education.

Brit, D. (2007). Impact of Globalization in Creating Sustainable Competitive Advantage. Retrieved June 25, 2017, from http://www.sdcexec.com/article/10289694/impact-of-globalization-in-creating-sustainable-competitive-advantage

Chen, G., Andrew, N. K., Goes, B. P., Gupta, A., & Marsden, J. R. (2006). Heuristics for selection robust database structures with dynamic query patterns. *European Journal of Operational Research*, 220–220.

Cohn, M. (2004). Writing Stories. *User Stories Applied for Agile Software Development*, 17–29.

Cohn, M. (2006). *Agile Estimating and Planning*. Upper Saddle River, New Jersey: Pearson Education.

Cohn, M. (2010). *Succeding with agile: Software development using scrum*. Upper Saddle River, New Jersey: Addison-Wesley.

Cram, W. A., & Marabelli, M. (2015). Have Your Cake and Eat it Too ? Managing Knowledge in Hybrid Agile-Traditional Development Projects. *Twenty-First Americas Conference on Information Systems, Puerto Rico*, 1–13.

Dahlman, C., & Edmund, A. (2007). *Technology, globalization, and international competitiveness: Challenges for developing countries*. *Industrial Development for the 21st Century*. http://doi.org/10.4016/12667.01

Dalling, T. (2017). *Programming for Beginners*. leanpub.

Davenport, T. H. (1998). Putting the enterprise into the enterprise system. *Harvard Business Review*, *76*(4), 121–131. http://doi.org/Technical Report

Dillon, C. (1999). Stretching toward enterprise flexibility with ERP. *American Production and Inventory Control Society: The Performance Advantage*, 38–43.

Dingsøyr, T., García, F., Ruiz, F., & Piattini, M. (2004). Deploying_Agile_Practices_in_Organizations_A_Case_Study, *3281*(November 2005), 79-90–90. http://doi.org/10.1007/b102170

Dingsoyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A Decade of Agile Methodologies: Towards Explaining Agile Software Development. *Journal of Systems and Software*, 1213–1221.

Doyle, M., Williams, L., Cohn, M., & Rubin, K. S. (2014). Agile Software Development in Practice, 32–45. http://doi.org/10.1007/978-3-319-06862-6_3

Dul, J., & Hak, T. (2008). *Case Study Methodology in Business Research* (1st ed.). Oxford: Butterworth-Heinemann.

Eldijk, J. v. (2016). *Agile ERP implementation methods: A Structured literature review*. Eindhoven.

Fair, J. (2012). Agile versus Waterfall: approach is right for my ERP project? In *PMI® Global Congress 2012—EMEA*. Marsailles: PA: Project Management Institute. Retrieved from http://www.pmi.org/learning/library/agile-versus-waterfall-approach-erp-project-6300

Fetouh, A. A., Abbassy, A. el, & Moawad, R. (2011). Applying Agile Approach in ERP Implementation. *IJCSNS International Journal of Computer Science and Network Security, 11*(8), 173–178.

Field, A. (2009). *Discovering Statistics using SPSS* (3rd ed.). London: SAGE Publications Ltd.

Finchman, robert G., & Scoot, A. M. (1999). An incremental process for software implentation, *40*(2).

Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising Agile Methods to Software Practices at Intel Shannon. *Uropean Journal of Information Systems*, 200–213.

Flackett, J. (2015). "How much does it cost to build a software application?" | Dr. John Flackett | Pulse | LinkedIn. Retrieved June 13, 2017, from https://www.linkedin.com/pulse/how-much-does-cost-build-software-application-dr-john-flackett

Gartner. (2016). *IT Key Metrics Data 2016: Key Applications Measures*. Retrieved from http://www.gartner.com/document/code/291414

Gibbert, M., Ruigrok, W., & Wicki, B. (2008). What passes as a rigorous case study? *Strategic Management Journal*, *29*(13), 1465–1474.

Gregor, S., & Hevner, A. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, *37*(2), 337–356.

Hagafors, R., & Brehmer, B. (1983). Does having to justify one's judgments change nature of the judgment process? *Organizational Behaviour and Human Decision Processes*, *31*(2), 223–232.

Harder, D. W. (2005). Extrapolation. Retrieved July 2, 2017, from https://ece.uwaterloo.ca/~dwharder/NumericalAnalysis/06LeastSquares/extrapolation/complete.html

Hevner, A., March, S., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75–105.

Hoffer, A. J. (2011). *Modern System Analysis and Design* (6th ed.). Upper Saddle River, New Jersey: Pearson Education.

Jørgensen, M. (2004). A review of studies on expert estimation of software development effort. *Journal of Systems and Software*, *70*(1–2), 37–60. http://doi.org/10.1016/S0164-1212(02)00156-5

Kalaimani, J. (2016). Implementing SAP with ASAP 8 Agile Methodology. *SAP Project Management Pitfalls*, *0*(0). http://doi.org/10.1007/978-1-4842-1389-6_2

Klaus, H., Rosemann, M., & Gable, G. (2000). What is ERP? *Information Systems Frontiers*, *2*(2), 141–162. http://doi.org/10.1023/A:1026543906354

Lawrence, R. (2009). Patterns for Splitting User Stories. Retrieved from http://agileforall.com/patterns-for-splitting-user-stories/

Lederer, A. L., & Prasad, J. (1992). Nine Management Guidelines For Better Cost Estimating. *Communications Of The ACM*, *35*(2), 51–59.

Lindvall, Basili, Boehm, Costa, Dangle, Shull, … Elkowitz. (2002). Empirical Findings in Agile Methods. *XP/Agile Universe 2002*, 197–2002.

McLaughlin, M. (2016). What Is Agile Methodology.

Ozcan Top, O., & Demirors, O. (2013). Assessment of agile maturity models: a multiple case study. In *Communications in Computer and Information Science 349* (pp. 130–141). Springer: Berlin Heidelberg.

Panorama Consulting Solutions. (2015). 2015 Erp Report. *Panorama Consulting Solutions*, 1–19. Retrieved from http://panorama-consulting.com/resource-center/2015-erp-report/

Pathfinder Solutions. (2012). *Effective TDD for Complex Embedded Systems*.

Pattion, J. (2008). The New User Story Backlog is a Map. Retrieved February 27, 2017, from http://jpattonassociates.com/the-new-backlog/

Patton, J. (2005). It's all in how you slice. *Better Software*, (January 2005), 16–40. Retrieved from http://www.stickyminds.com

Peffers, K., Tuunanen, T., & Gengler, C. E. (2006). The design science research process: a model for producing and presenting information systems research. In *Proceedings Design Research Information Systems and Technology DESRIST'06* (p. 83–106.).

Pikkarainen, M., & Passoja, U. (2006). An Approach for Assessing Suitability of Agile Solutions: A Case Study. In *International conference of eXtreme Programming and agile process in software engineering* (pp. 18–23). http://doi.org/10.1007/11499053

Pressman, R. S. (2009). *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. http://doi.org/10.1017/CBO9781107415324.004

Qumer, A., & Henderson-Sellers, B. (2008). A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of Systems and Software*, *81*(11), 1899–1919. http://doi.org/10.1016/j.jss.2007.12.806

Ruël, H. J. M., Bondarouk, T., & Smink, S. (2012). The waterfall approach and requirement uncertainty: an in dept case study of an enterprise system implementation at a major airline company. In *Project Management Techniques and Innovations in Information Technology* (pp. 49–83).

Salo, O. (2004). Improving Software Process in Agile Software Development Projects: Results from Two XP Case Studies Outi. In *Proceedings of the 30th EUROMICRO Conference*.

Salo, O. (2005). Systematical Validation of Learning, 106–110.

Schatz, B., & Abdelshafi, I. (2005). Primavera gets Agile: A successful transition to Agile development. *IEEE Software*, *22*(3), 36–42. http://doi.org/10.1109/MS.2005.74

Shankarnarayanan, S. (2000). ERP systems—using IT to gain a competitive advantage. Retrieved from http://www.angelfire.com/co/troyc/advant.html

Sidky, A., Arthur, J., & Bohner, S. (2007). A Structured Approach to Adopting Agile Practices : The Agile Adoption Framework A Structured Approach to Adopting Agile Practices : The Agile Adoption Framework. *Innovations in Systems and Software Engineering*, *3*(3), 203–216.

Sinain, P. (2014). SurveyLegend | What are Likert-Type Scale Responses. Retrieved June 13, 2017, from https://www.surveylegend.com/likert-type-scale-responses-examples-with-examples/

Sindos, A. T. (2001). Evaluating the ERP Implementation Project. *Juropean Journal of Information Systems.*

Sowan, I., & Tahboub, R. (2015). Erp Systems Critical Success Factors. *International Journal of Advanced Computer Science and Applications*, *6*(6), 191–196.

Stender, M. (2002). Outline of an Agile Incremental Implementation Methodology for Enterprise Systems. *Proceedings of the Eighth Americas Conference on Information Systems*, 907–917.

Sureshchandra, K., & Shrinivasavadhani, J. (2008). Moving from waterfall to agile. *Proceedings - Agile 2008 Conference*, 97–101. http://doi.org/10.1109/Agile.2008.49

Wiegers, K., & Beatty, J. (2013). *Software Requirements*. Retrieved from https://books.google.com/books?hl=en&lr=&id=nbpCAwAAQBAJ&pgis=1%5Cnhttp://www.amazon.com/dp/0735679665/ref=pe_385040_30332190_TE_M3T1_ST1_dp_3

Zughoul, B. (2016). Software Process Models and Enterprise Resource Planning System. *International Journal of Advanced Research in Computer and Communication Engineering*, *5*(5), 489–494. http://doi.org/10.17148/IJARCCE.2016.55126

10 Appendixes

10.1 Appendix A: brief intro into agile methods

10.1.1 Extreme Programming (XP)

Extreme Programming is probably the most used approach to agile software Development and was first proposed by Kent Beck (Beck, 2004). This approach contains the following phases: planning, design, coding, and testing (Figure 1).

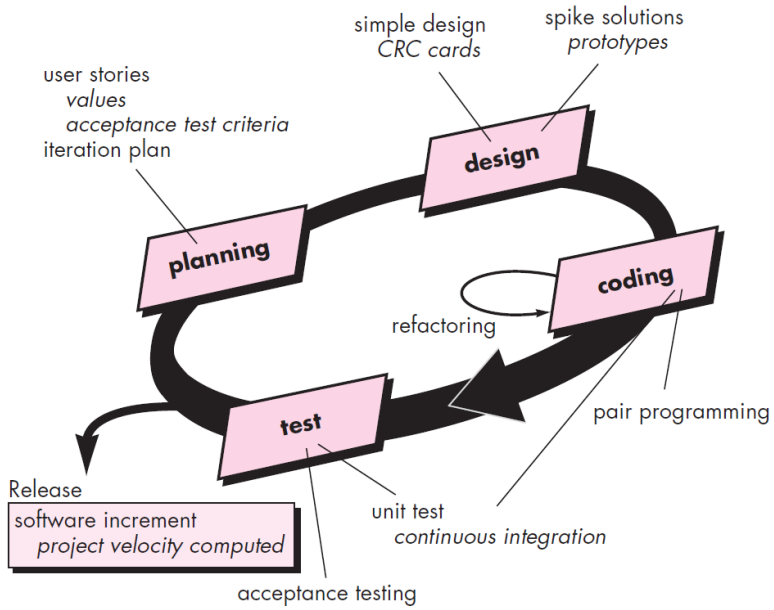


Figure 1: Life cycle of the XP process. (Pressman, 2010).

10.1.1.1 *Planning*

In the planning phase (also called *the planning game*) requirements are gathered that enables the XP team to understand the business context for the software and to get a broad feel for required output and major features and functionality. User stories (similar to use cases) are written by the customer and a value based on the business value of the feature or function is assigned to them. The XP team asses each story and estimate the effort each story requires. The stories are then prioritized and a schedule is agreed upon. Customers and developers break the schedule up into a number of iterations that will implement a set of user stories. Stories with a higher value or higher risk are implemented first. After the first release, the XP team computes the number of stories that they implemented. This is then used to estimate deliver dates and schedule subsequent releases. It can also be used to check the projects feasibility. During development new stories can be added and old stories can be changed or removed. The XP team modifies their plans accordingly.

10.1.1.2 *Design*

The Design phase provides implementation guidance for a story as it is written. A simple design is always preferred over a more complex representation. If a difficult design problem is encountered as part of the design of a story, XP recommends the immediate creation of an operational prototype of that portion of the design. Called a spike solution, the design prototype is implemented and evaluated. Design occurs both before and after coding commences. Another important notion in XP is that of refactoring, which happens continually. The thought behind this is that the construction activity itself will provide the XP team with guidance on how to improve the design.

10.1.1.3 *Coding*

After design coding is not immediately started, instead a set of unit tests are created for the stories that will be implemented in the current increment. This helps developer's focus on implementing what is important to pass the test. Once a piece of functionality is completed it must be tested immediately, so that it can instantly provide feedback to the developers. Completed work is continuously integrated with the other work in order to avoid compatibility and interfacing problem and to uncover errors early. Coding is done by two people work together at one computer workstation. This practice is called **pair programming** and enables real time problem solving, quality assurance, keeps the developers focused on the problem at hand.

10.1.1.4 *Testing*

Unit tests that are created should be implemented using a framework that enables them to be executed easily and repeated. This way integration and validation testing of the system can occur on a daily basis and provides the XP team with an indication of progress. It also raises warning flags early if things go wrong. Next to unit test there are acceptance tests, also called customer tests. These tests are specified by the customer and focus on overall system features and functionality that are visible and reviewable by the customer. Acceptance tests are based on the user stories that have been implemented in one of the iterations.

10.1.2 Scrum

Scrum does not focus on specific software development techniques. Instead, it concentrates on how team members should function to achieve system flexibility in a changing environment. Scrum includes three phases: pre-game, development and post-game. A visual representation of scrum is given in figure 2 down below.

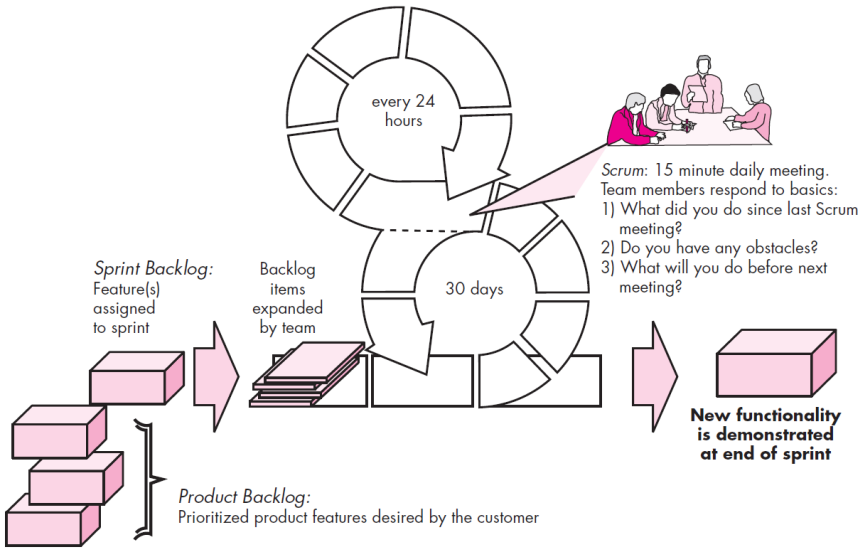


Figure 2: Scrum process flow. Source (Pressman, 2010)

10.1.2.1 *Pre-game phase*

In the **pre-game phase** the product **backlog** is created and a **planning poker** game is held to estimation the efforts needed to implement each on the list. The backlog is constantly updated with new and more detailed items. This phase also includes: the definition of the project team, tools and other resources, risks assessment and control and training needs. After an iteration (sprint) the backlog is updated and reviewed by the team and a commitment is made on the next iteration.

10.1.2.2 *Game phase*

The **game phase** in scrum is treated as a black box where the unpredictable is expected. The different relevant environmental and technical variables are identified, observed and controlled through various scrum practices applied in a **sprint.**

A sprint tries to implement a set of requirements from the backlog in a predefined **time box**, which typically last 30 days. At the start of the sprint the **definition of done** is defined to enable a good communication. During a sprint no changes are introduced, allowing the team to work in a stable environment. In a sprint **daily meetings** are held to help the team to uncover potential problems early and enable "knowledge socialization". A **task board** is also maintained and **backlog grooming** is performed regularly. After the time box of a sprint is elapsed a **sprint review meeting** is held. In this meeting a demonstration of those functionalities that are finished, according to the predefined definition of done, is given to all stakeholders. This demonstration is the used access the performed sprint and to make decisions about the following activities.

10.1.2.3 *Post-game phase*

The **Post-game phase** is entered when there it is agreed upon that no new items or issues can be found that should be on the backlog. This means that the system is ready for release, which is prepared in this phase. This includes tasks as integration, system testing and documentation.

The scrum team consists of a scrum master, product owner, the scrum team, the customer and management. It is suggested that the team should comprise of five to nine project members. In case more people are available multiple teams should be formed.

In the team the scrum master ensures that the project follows the scrum practices and makes changes to the process where necessary. The product owner is the person who is officially responsible for the project, managing, controlling it and making the product backlog visible. He makes the final decision on task related to implementing the backlog items and translates issues from the backlog into features that need to be implemented. The scrum team is a group of people with the authority to organize it-self and determine the actions necessary to achieve the goals of a sprint. The customer helps with defining the items on the product backlog. Management makes the final decisions, enforces certain standards and practices and participates in the setting of goals and requirements.

10.2 Appendix B: Screenshot of issue and project tracking tool

10.2.1 Task board

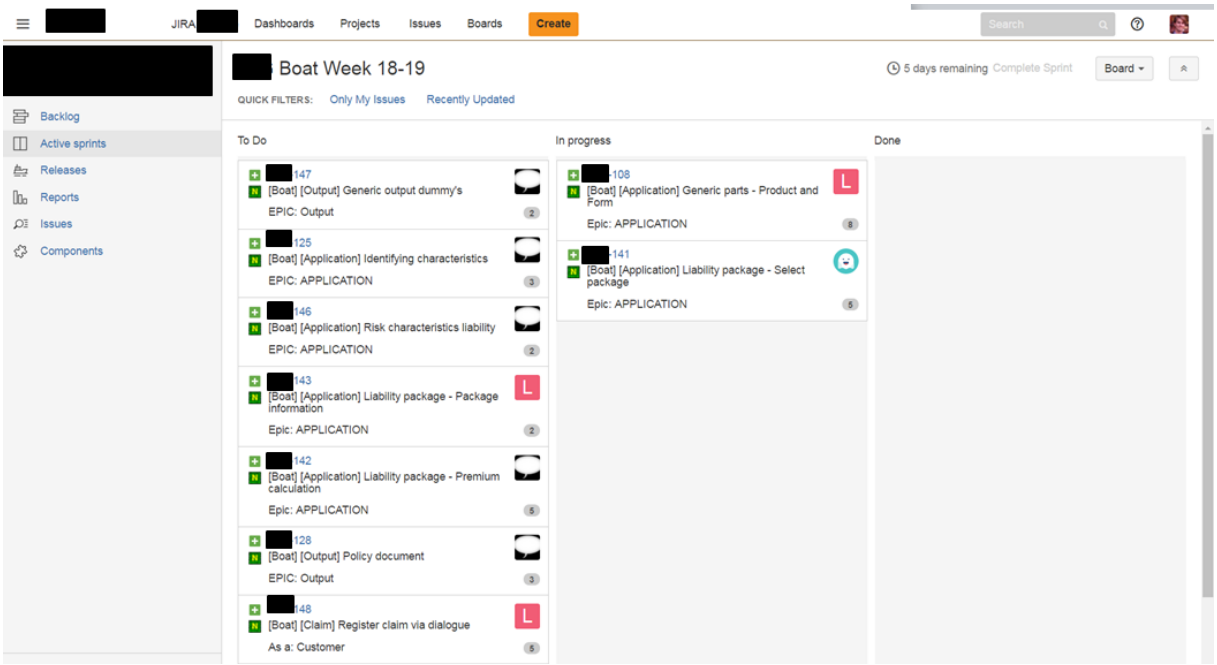
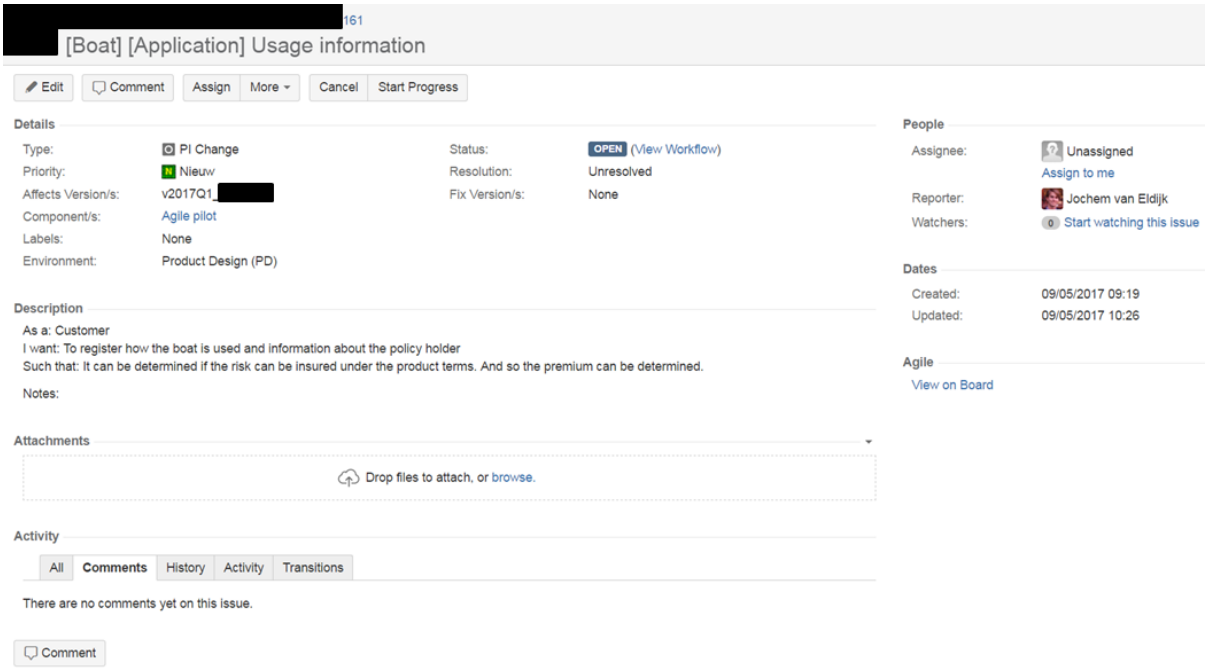
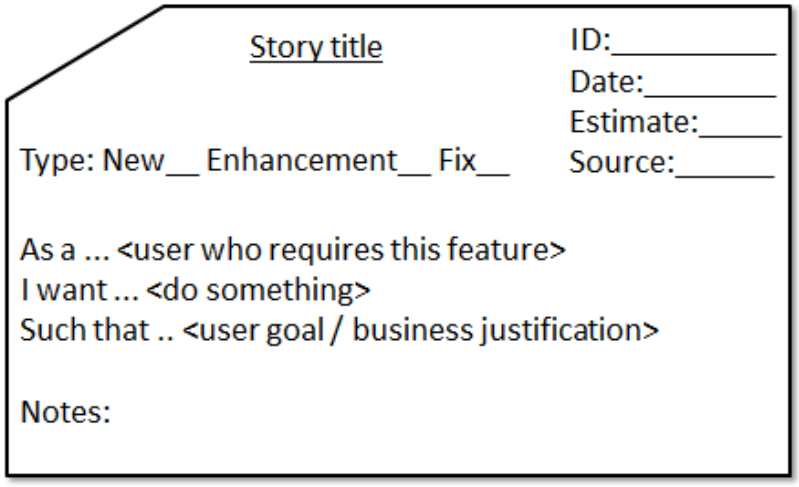


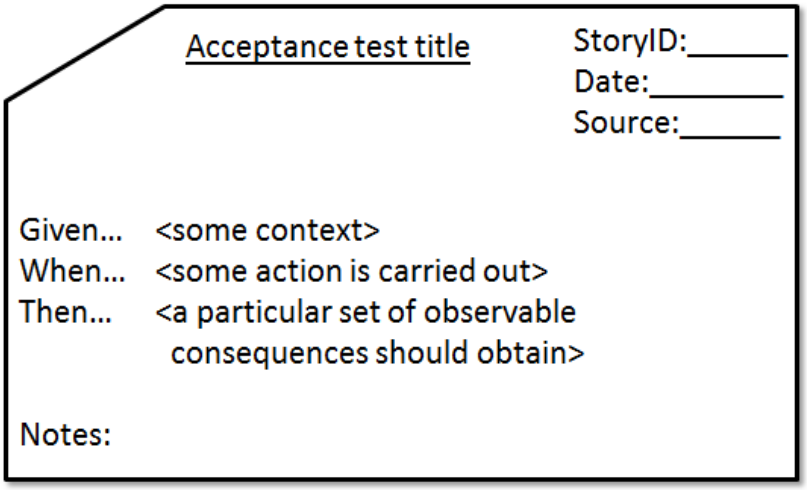
Figure 17: task board sprint 1

10.2.2 User story



10.3 Appendix C: User story and acceptance test templates





10.4 Appendix D: Project charter

This project charter

- Focuses the project on what needs to be done and why it needs to be done
- introduces the project to other stakeholders

Agreed upon team values and norms

- Listen to everyones thoughts
- Live up to commitments
- Share knowledge
- Response quickly (within 4 hours) on the fixed days
- Don't be afraid to ask someones opinion or to give your own
- Take responsibilit and stand by decisions
- Actively participate
- Have fun!

Elevator pitch of the project (1/2)

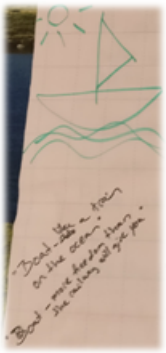
- Increase usability
- See the total process and whole pciture
- Combine team effort increaing underanding and commitment amongst each other, ultamitely leading to a better product
- Focus on business value

Elevator pitch of the project (2/2)

- Closer collaboration
- Pilot which we hope can change working methods and product maintenance
- See the real (working) product quicker
- Everybody has the same goal and sees the same picture

product box

- Good price
- Simple to understand
- Good insurance / cover
- No excess in case the boat has a tracking system
- STP – quick, easy, do it yourself
- We help you to prevent damage



Project success criteria

- Business understads what we build, why and how it works
- Max. 5 enhancements are needed
- Simplify process and product
- Web and STP – less time used / easy to understand

10.5 Appendix E: retrospective improvement topics

10.6 Appendix F: Action point list

Improvement topic: Schedule				
Finding	Improvement action	Actor	Validation plan	validation
Difficulties with finding time to spend on the product	Use deadlines on Jira tickets	Jochem	check if all Jira tickets have a deadline	No deadlines were added, however tickets were picked up early in the sprint.
Questions were only answered late. Blocking the work on some tickets	try to answer same day	Jochem	Monitor response times	Some questions were answered within one day; others took multiple days
People started working on user stories at the end of the sprint.	The company has to start on tickets early in the sprint; assign user stories on first working day after planning meeting.	Joris	Check Jira the firs working day after the planning meeting	

Improvement topic: Information exchange				
Finding	Improvement action	Actor	Validation plan	validation
The costumer felt like they could have provided information on some of the user stories earlier in the sprint.	Insurer tries to add information as soon as possible	Hanne	At least 4 user stories got additional information from insurer, before someone asked for it.	
Not all insurer members had access to Jira, therefore missing information on user stories and felling excluded from the project	All insurer team members get access to Jira; this should improve how much time we spend on the product outside of the sessions.	Jochem	Check Jira if all insurer members have access	All members were granted access
Some questions miss some context information, causing people to misinterpret each other	Team members can make mini designs so it is easier to know what it has to look like in the dialogue; use other product / make example in Word	All	Discuss in next retrospective if this has improved	
Some tickets have to little information	We make a comment in the Jira ticket whenever we provide information for the tickets (email, attachment, acceptance test)	All	Check comments in Jira if all relevant information is referred to.	

Figure 18: action point list of retrospective one

10.7 Appendix G: Semi structured interviews for AIM evaluation

Opening of Interview

Thank you for agreeing to be interviewed for our research on an agile implementation method. We want to find out about your experience with applying agile practices within a EIS implementation. We are collecting information so that we can evaluate the applicability and effects that these practices have. The interview should take about 60 minutes and will be recorded for study later on. Any information that you provide will be anonymized and shall not be used to evaluate your performance as a person in any way. Are you still happy to proceed with the interview?

10.7.1 Demographics

Let's start with some information that will help me describe your characteristics:

Interviewer: How old are you?
Interviewer: Note if the respondent is male or female: M | F
Interviewer: For which company do you work?
Interviewer: What is your official job title?
Interviewer: What is your role within the Boat project
Interviewer: Have you any experience with EIS implementations?
Interviewer: how many years?
Interviewer: how many project?
Interviewer: Have you any experience with agile development methods?
Interviewer: how many years?
Interviewer: how many project?
Interviewer: What was your attitude towards an agile implementation method before this pilot?
Interviewer: Has this changed in the course of the project?
Interviewer: what has changed way and why?

10.7.2 Agile practices

Next up are some questions about the agile practices that were applied during the project. For each practice I will ask 3 questions. But first, I will provide a short description of the practice in question.

Project **CHARTER**

The project charter is a high-level summary of the project's key success factors that was created during the exploration meeting on a flipchart-sized sheet of paper.

Interviewer: What was your experience with this practice in the project?
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
Interviewer: How do you think that this practice affected the outcome of the project?

Planning game

The planning game is the practice of determining which user stories should be implemented in the next sprint by assigning priorities to them.

Interviewer: What was your experience with this practice in the project?
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
Interviewer: How do you think that this practice affected the outcome of the project?

Product backlog

The product backlog is the list of user stories which at a given moment, are known to be necessary and sufficient to complete a project or a release. In the pilot the product backlog was shown in Jira and held the user stories that still needed to be assigned to a sprint

Interviewer: What was your experience with this practice in the project?
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
Interviewer: How do you think that this practice affected the outcome of the project?

Sprint

A sprint is development cycle of fixed length, at the end of which a demonstrable piece of functionality is delivered.

Interviewer: What was your experience with this practice in the project?

Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
Interviewer: How do you think that this practice affected the outcome of the project?

Daily meeting (company only)

During daily meeting each team members briefly describe any "completed" contributions, any obstacles that stand in their way and their goals for that day

Interviewer: What was your experience with this practice in the project?
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
Interviewer: How do you think that this practice affected the outcome of the project?

Sprint review

At the end of each sprint, a sprint review meeting is held. During this meeting, the Scrum team shows what they accomplished during the sprint. Typically this takes the form of a demo of the new features.

Interviewer: What was your experience with this practice in the project?
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
Interviewer: How do you think that this practice affected the outcome of the project?

task board

The Task board is a board that displays the status of user stories as either: “to do”, “in progress”, “ready for test” or “done”. This board was visible in Jira during the pilot project and showed the user stories that were assigned to a sprint.

Interviewer: What was your experience with this practice in the project?
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
Interviewer: How do you think that this practice affected the outcome of the project?

Sprint retrospective

The Sprint retrospective is a meeting in which the team reflects on the most significant events that have occurred in that sprint, and take decisions aiming at remediation or improvement

Interviewer: What was your experience with this practice in the project?
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
Interviewer: How do you think that this practice affected the outcome of the project?

Burndown cart (only for manager)

A burndown chart is a graph which shows the quantity of work remaining and the time elapsed since the start of the project

Interviewer: did you apply this practice? If not why?
Interviewer: Did you face any challenges when applying this practice?

User stories

User stories are requirements that are written on a physical card. These requirements are a functional piece of the product that provides value to the customer and can be implemented independently from each other. This practice helps split the work in small pieces

Interviewer: What was your experience with this practice in the project?
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
Interviewer: How do you think that this practice affected the outcome of the project?

Acceptance testing

An acceptance test is a description of the behavior of a software product, expressed as a usage scenario. Executing the acceptance test generally result in either a pass or fail

Interviewer: What was your experience with this practice in the project?
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
Interviewer: How do you think that this practice affected the outcome of the project?

Story points

Story points is a measure used to indicate the size of a user story relative to one another

Interviewer: (manager question) How did you experience the estimation of story points?
Interviewer: What is your opinion on using story points as an indication of the size of a user story

Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
Interviewer: How do you think that this practice affected the outcome of the project?

Velocity

The velocity of a team is the amount of user stories that were completed during that sprint expressed in the units used for effort estimates

Interviewer: What was your experience with this practice in the project?
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
Interviewer: How do you think that this practice affected the outcome of the project?

10.7.3 Project outcomes

That was the final question about agile practices. Next up are some questions that relate to the projects performance measures. These performance measures are grouped in tree categories. First, I will ask some questions about Team related performance attributes. This is followed by some questions about the quality of the product. And I will end with some questions about the implementation process.

10.7.3.1 *Team related outcomes*

productivity

Interviewer: Compared to previous project you participated in, how was the team's **productivity**?
Was this for example lower or higher than other project you participated in ?
What do you think is the cause of this difference?

responsibility

Interviewer: Compared to previous project you participated in, how was the team's **responsibility**?
Was this for example lower or higher than other project you participated in ?
What do you think is the cause of this difference?

motivation

Interviewer: Compared to previous project you participated in, how was the team's **motivation**?

Was this for example lower or higher than other project you participated in ?
What do you think is the cause of this difference?

Knowledge

Interviewer: Compared to previous project you participated in, how was the team's **Knowledge sharing**?
Was this for example lower or higher than other project you participated in ?
What do you think is the cause of this difference?

collaboration

Interviewer: Compared to previous project you participated in, how was the team's **collaboration**
Was this for example lower or higher than other project you participated in ?
What do you think is the cause of this difference?

10.7.3.2 *Product related outcomes*

quality

Interviewer: Compared to previous project you participated in, What do you think about the **Product quality**?
Interviewer: Is this lower or higher than other project you participated in ?
Interviewer: What do you think is the cause of this difference?

usability

Interviewer: Compared to previous project you participated in, What do you think about the **Product usability**?
Interviewer: Is this lower or higher than other project you participated in ?
Interviewer: What do you think is the cause of this difference?

maintainability

Interviewer: Compared to previous project you participated in, What do you think about the **Product maintainability**?
Interviewer: Is this lower or higher than other project you participated in ?

Interviewer: What do you think is the cause of this difference?

bugs / defects

Interviewer: Compared to previous project you participated in, What do you think about the **Product bugs / defects**?
Interviewer: Is this lower or higher than other project you participated in ?
Interviewer: What do you think is the cause of this difference?

satisfaction

Interviewer: Compared to previous project you participated in, What do you think about the **customer satisfaction**?
Interviewer: Is this lower or higher than other project you participated in ?
Interviewer: What do you think is the cause of this difference?

satisfied

Interviewer: (client) Compared to previous project you participated in how **satisfied** are you with the
Interviewer: Is this lower or higher than other project you participated in ?
Interviewer: What do you think is the cause of this difference?

10.7.3.3 *process related outcomes*

understandability

Interviewer: Compared to previous projects you participated in how was the **understandability** of the implementation process
Interviewer: Is this lower or higher than other project you participated in ?
Interviewer: What do you think is the cause of this difference?

flexibility

Interviewer: Compared to previous projects you participated in how was the **flexibility** of the implementation process
Interviewer: Is this lower or higher than other project you participated in ?
Interviewer: What do you think is the cause of this difference?

transparency

Interviewer: Compared to previous projects you participated in how was the **transparency** of the implementation process
Interviewer: Is this lower or higher than other project you participated in ?
Interviewer: What do you think is the cause of this difference?

10.7.4 Next steps company

Finally I have some questions regarding the application of agile within the company

Recommendation

Interviewer: Would you recommend the company to adopt agile as its main implementation method? Why would you do that? If yes, which parts should be paid additional attention to?
Interviewer: Do you think this method would in larger projects? Why? / why not? Which parts should be paid additional attention to?
Interviewer: Do you think agile could be used by multiple teams working in parallel? Why? / why not? Which parts should be paid additional attention to?

10.8 Appendix H: interview transcription

10.8.1 Interview one

Opening

Thank you for agreeing to be interviewed for our research on an agile implementation method. We want to find out about your experience with applying agile practices within a EIS implementation. We are collecting information so that we can evaluate the applicability and effects that these practices have. The interview should take about 60 minutes and will be recorded for study later on. Any information that you provide will be anonymized and shall not be used to evaluate your performance as a person in any way. Are you still happy to proceed with the interview?

10.8.1.1 *Demographics*

Let's start with some information that will help me describe your characteristics:

Interviewer: How old are you?
Hired consultant insurer: 32

Interviewer: Note if the respondent is male or female: Female

Interviewer: For which company do you work?
transcendent group

Interviewer: What is your official job title?
Senior consultant

Interviewer: What is your role within the Boat project
Representative for insurer

Interviewer: Have you any experience with EIS implementations?
It is the first time I have worked on an insurance systems implementation, but I have experience in other sectors

Interviewer: how many years?
7 years

Interviewer: how many project?
A lot, I cannot give you an exact number

Interviewer: Have you any experience with agile development methods?
No this was the first time, yep first time

Interviewer: What was your attitude towards an agile implementation method before this pilot?
Positive

Interviewer: Has this changed in the course of the project?
No this has not changed

10.8.1.2 *Agile practices*

Next up are some questions about the agile practices that were applied during the project. For each practice I will ask 3 questions. But first, I will provide a short description of the practice in question.

Project charter

The project charter is a high-level summary of the project's key success factors that was created during the exploration meeting on a flipchart-sized sheet of paper.

Interviewer: What was your experience with this practice in the project?

it was positive I think it was good that it was conducted together by insurer and the company

Interviewer: Did you face any challenges when applying this practice?

no I don't think so, of course we have been working together with the people of the company for over a year now so that help to get to a common understanding

Interviewer: How do you think that this practice affected the outcome of the project?

I think it was good to have a common understanding from the beginning of the project. but we had to have an internal meeting after that meeting to make sure everyone understood the agile way of working

Interviewer: and do you think it is helpful?

Yes

Interviewer: why do you think that, what would be the main reason?

Well we have the common understanding of the projects, how we work together, how we will interact. We also did the value I'm not sure how much that helped, but I think the common understanding and the cooperation is good.

Planning game

The planning game is the practice of determining which user stories should be implemented in the next sprint by assigning priorities to them.

Interviewer: What was your experience with this practice in the project?

It was good, it was very good. It was good for us to prioritize wat we wanted to have first and wat comes second and to see how we can have party of the product first. So it was good. It was a bid unclear maybe to the product owner at insurer a new way of think in in term of agile I think it worked well

Interviewer: Did you face any challenges when applying this practice?

In our Internal meeting we also worked on user stories to come, so we could work a bit ahead. So we can see what we prioritize it is not that we don't mention it and it is not important, but it just comes latter. So that was good to get all the team members on board. To see the complete product in the whole project

Interviewer: so it needed a bid of practice?

yes

Interviewer: How do you think that this practice affected the outcome of the project?

Yes I think it is very good so that we don't always need to include everything in the product, but that we can focus on the most important things and also for me it very important to try and take the costumer view.

Interviewer: And why would that be important?

Quite often we focus on the internal view, wat is important to the internal users, but in the end this product is for our customers so we need to make sure that they get all the information

Interviewer: so do you think this positively or negatively influenced to project outcome?

Positively

Product backlog

The product backlog is the list of user stories which at a given moment, are known to be necessary and sufficient to complete a project or a release. In the pilot the product backlog was shown in Jira and held the user stories that still needed to be assigned to a sprint

Interviewer: What was your experience with this practice in the project?

I think it was quite important to see the totality of the product and to kind and visualize that there is

more to come and what we will later on prioritize. But it's good to have it in the backlog and to see what is coming. I think it was a bit difficult in the first sprint, since not all of us had access to Jira, to see what was in the backlog and what was discussed and not. But once we had access we could create tickets there to start working on.

Interviewer: Did you face any challenges when applying this practice?

No it is a bid new to quite user stories in itself, to have the right amount of information in there and the details of information. Since we were user to working in a different way. But it was good to…

Interviewer: but creating them in the backlog and placing them there that was not difficult?

No that was good! And the combination of our priorities together with the company assessment of kind of the points, well it worked.

Interviewer: How do you think that this practice affected the outcome of the project?

Very positive, well as I mentioned it is good to see in each sprint there are fewer user stories. But then it helps to get a bit clearer picture a bit earlier and to see the reason why we prioritize some things and why others are in the backlog. It helps with the complete overview and complete understanding of the whole team. To see what it coming more lately and to see what we need to work on and what information we need to collect.

Sprint

A sprint is development cycle of fixed length, at the end of which a demonstrable piece of functionality is delivered.

Interviewer: What was your experience with this practice in the project?

I thought it was really good to have a product demo so quickly after the first sprint. on the first sprint that we actually good to see the product demonstrated in the test environment

Interviewer: yeah that is the demo practice. But the sprint is actually just the time that it's in a short period.

Yeah it was good. It was maybe more challenging to find time and prioritize working on the boat product working in between the sprint meetings. But yeah I think it was good to work within a short period of time.

Interviewer and why do you think that is good?

Because we work a bid more concentrated on the whole product instead of having a very long period of time were we first work on the dialoged and then on the output and then on the claims. Now we could go much quicker on the whole product.

Interviewer: Did you face any challenges when applying this practice?

well it was more to find the time in between the sprint meeting to work on it since we were very quiet busy with other workshops and other products. and also kind of the first sprint we had to take

some time to reflect on what was good and not and to kind of have a common understanding that we cannot all work on the Jira tickets at the end of the sprint period. In order to move forward we need to provide information early and to answer question early in the sprint period. If everyone wait to work in the last minute then we don't move forward.

Interviewer: How do you think that this practice affected the outcome of the project?

I think it was a positive of course I must say that since we had. This is one of the last product we work on. It that also affected the outcome of it because we are so used to wat is generic and hat is product design so we know the general processes and what the dialogs will look like in mind. So it helped very much when we worked on the boat product in the agile way. Couse we could go quite fast to the solution, without having the discussion we had on the other products.

Interviewer: and without that experience?

then I think it would be more difficult but maybe you would reather need to spend more time on planning or to be able to deddicate more time to the product itself. in the sprint

Sprint review

At the end of each sprint, a sprint review meeting is held. During this meeting, the Scrum team shows what they accomplished during the sprint. Typically this takes the form of a demo of the new features.

Interviewer: What was your experience with this practice in the project?

Very good, that was the most postive of the whole agile way of working.

Interviewer: and why is it positive?

Because it is so positive to see the product in a short amount of time.

Interviewer: and except that it is of course fun to see working product does it also help?

Yeah it helps a lot to kind of see. Well first of all you see that it is actually working. We have a product. Even though it just contains a few user stories. And this is something you could present to a costumer and start selling right away. So in terms of costumer view it works. And it also helps to kind of visualize the product instead of seeing it just on a word document.

interviewer: does it also help with comming up with new ideas? or seeing problems in the product?

Hump, yes but I think maybe that the scope of the project / prduct was not kind of completely define a new product. That was not so much in scope. We have tried to simplify it. And it was good to see that that works. To see that we do not need that many question. That the product works with simplification. But in term of new features we focus less on that. But it is ones of the things. For example the super package is something for, not the next sprint, but for the last sprint. As improvement.

Interviewer: Did you face any challenges when applying this practice?

I think there were some minor issues. Access to the test environment and the user role for testing.

But apart from that it was fine.

Interviewer: How do you think that this practice affected the outcome of the project?

Very positive. I think it was helpful for the whole team to see the whole product so quickly and it is a lot easier to priories the user stories for the next sprint when you have the product. And I also thought it was quite / very impressive of the company that they managed to work so quickly to get to product on air.

Ow and one more thing. It also helped us prioritize our work when we worked a lot with reviewing word document we would commend a lot on wordings and minor details of the wording of what the coverages include, typos and things like that. But when we worked in this agile way we focused and got to see the product. We did not spend so much time on the wording in a word document. So it was the most positive.

Interviewer: and that is due to the demo or is it for agile in general.

Yeah both. The demo because you then see the product and we see the coverages there, so it is not that we need a good explanation of what the coverages contain in English. And also the agile way of working.

Task board

The Task board is a board that displays the status of user stories as either: “to do”, “in progress”, “ready for test” or “done”. This board was visible in Jira during the pilot project and showed the user stories that were assigned to a sprint.

Interviewer: What was your experience with this practice in the project?

I took us quite a lot of time to get access to Jira, so I only saw it last Wednesday. So we had already passed a sprint and a held. That was a bid a problem. But then when I got access it really helped to see, where we are and what are we working on, which tickets. We have now several different Jira pages per project. This page was a bid different than the one we used to working on. So it was a bid more a the company version of the board. but it helped to show what needed to be tested and what we needed to provide information on and when provided information on the to do that the company quickly responded, and I think that information of the test results, which was good. Other Jira pages.

Interviewer: Did you face any challenges when applying this practice?

I had some minor technical problems with the acceptance moving from one swim lane to another which was a bid different from to other Jira page

Interviewer: How do you think that this practice affected the outcome of the project?

A bid negative that we got access to Jira so late. Because when I got it I has to quickly use that date of working. Because for user stories I wrote were on a piece of paper so I did not see it in the system. So that was a bid negatively. but I see that when I got access it was positive

Sprint retrospective

The Sprint retrospective is a meeting in which the team reflects on the most significant events that have occurred in that sprint, and take decisions aiming at remediation or improvement

Interviewer: What was your experience with this practice in the project?

It took a lot of time I thought. Maybe too much to me. But it was good to get everyone's opinion and make action points and easy and simple solutions to improve. It was nice to give positive feedback to each other. Which we do not do so often. Usually we just commend on things we want to change.

Interviewer: and do you think the improvements are also carried out?

Yeah because people can indicate what they think is important.

Interviewer: Did you face any challenges when applying this practice?

It did not take that much time, but relative to the small meeting it took much time. For the rest it was good.

Interviewer: How do you think that this practice affected the outcome of the project?

Positive because it helped the people work together and coordinate the work and to make sure that we provide the information in a timely matter.

User stories

User stories are requirements that are written on a physical card. These requirements are a functional piece of the product that provides value to the customer and can be implemented independently from each other. This practice helps split the work in small pieces

Interviewer: What was your experience with this practice in the project?

Positive. It was kind of new way of working for me. It was very positive that we could split it down to small pieces and then the company could indicate how small things could be and what we could group together. I had to get used to working with user stories and acceptance tests.

Interviewer: it required a bit of learning / getting used to.

Yeah. Also the amount of detailed information and how much information we should provide in the user stories.

Interviewer: Did you face any challenges when applying this practice?

in some areas, for example in the word document we kind of just write acceptance rules, in some areas I felt like writing user stories took more time. For small user stories. Because when we also had internal meeting user stories were quite quick and I kind of understood what we meant. But of course if it was not enough information provided for the company... but in total I think it was... yeah positive to divide the work out in smaller pieces and then to prioritize.

Interviewer: How do you think that this practice affected the outcome of the project?

I think positively. But maybe we could have spent more time on explaining the user stories and the ay of information required. It was good to have review of a user stories to see how to provide more information. So maybe go through to it a bit more in the planning meeting.

Acceptance testing

An acceptance test is a description of the behavior of a software product, expressed as a usage scenario. Executing the acceptance test generally result in either a pass or fail

Interviewer: What was your experience with this practice in the project?

Well... it was maybe a bit difficult to know or kind of test the dialogue in the whole. Not partly of each user story. Not split it as much maybe. According to the user story. And I also did not know about the accept in reject bottom. But I could test the dialogue and enter the results of the testing. I just did not do the test and reject. Or I write it more in the comments.

Interviewer: you mean in the Jira tool?

yea

Interviewer: but did the descriptions of the AT help with the testing? Or with clarifying with what a user story should do or should be?

Yeah well, yes. But this is a bid very different of me. I try to test the whole process of entering the dialogue and entering the application. Whereas the acceptance test are quite technical for the company. Were you just can enter a dialogue or present a premium. For me it is more appropriate to test the end to end process.

Interviewer: so it was a bid to small?

Yeah.

Interviewer: so that was also the challenge that you faced with this practice?

Yeah maybe for example with regards to the claim you had to enter a claim and if you are not allowed to save it. Then it does not matter if you can enter it because it is not saved. So for me it is more important to see the end to end and have the costumer perspective, because it is not just the first technical thing to present it on a screen.

Interviewer: How do you think that this practice affected the outcome of the project?

It was good that more members of the team could do the AT when to get access to the environment and Jira. And maybe we did not completely follow the acceptance test.

Interviewer: so you did not think it added any value? You did not really use it?

No I did you it. It was good that we could do the test. I just think we deviated a bid form the test

description.

Interviewer: so it helped a little?

Yes. But when we did the user stories the test description was a bid 1 to 1 to the user stories. But maybe we can write the user stories and then have 1 to 2 acceptance test that can do the process of the whole sprint.

Story points

Story points is a measure used to indicate the size of a user story relative to one another

Interviewer: What is your opinion on using story points as an indication of the size of a user story

It was a bid unclear to me in the beginning because we did not have access to Jira. So I did not see them and I forgot a bid about them. But now this print and this meeting today the manager explained a bid more about those points and we could really follow their discussion about the point and it was good to see their evaluation of giving points. And then we could measure that up to our prioritization so then it was very positive. So it was more to have time to go through it together and see it as well.

Interviewer: so it helped with the planning?

Yes but a bid later in the process. It is more important now that the project is coming to a close so that we can prioritize in the time that we have left.

Interviewer: so you are saying that in the start it is not that important, because everything needs to be implemented any way. And in the end it is more important to see for the prioritization

yes

Interviewer: How do you think that this practice affected the outcome of the project?

Positive.

Interviewer: and why do you think that?

Because it helps us to prioritize.

interviewer: could you give me an example?

For example the super package, which we did not have today. And the company indicated that it had many points. And that makes us see: do we want a complete working product or do we want new features. For us a working product was more important. By going through the points together we could also alter the user stories, which effected the amount of points. This helped simplify the solution / product.

Velocity

The velocity of a team is the amount of user stories that were completed during that sprint expressed in the units used for effort estimates

Interviewer: What was your experience with this practice in the project?

Fine. Although it required a bit of learning. It helped with the companies explanation of what was a good level / amount of user stories that could be implemented in a sprint. And that helped us in the prioritization.

10.8.1.3 *Project outcomes*

That was the final question about agile practices. Next up are some questions that relate to the projects performance measures. These performance measures are grouped in three categories. First, I will ask some questions about Team related performance attributes. This is followed by some questions about the quality of the product. And I will end with some questions about the implementation process.

Team related outcomes

Productivity

Interviewer: Compared to previous project you participated in, how was the team's **productivity**?

Higher than other projects. we managed to produce a lot more in a shorter time

Interviewer: What do you think is the cause of this difference?

Because we had little time we were forced to focus on what was really important and go straight to the solution in the short amount of time that we had.

Interviewer: so it helped make clear that time is limited?

Yeas and we also spend less time on reviewing and documentation.

Responsibility

Interviewer: Compared to previous project you participated in, how was the team's sense of **responsibility**?

Varying. Managers did very well. Prioritizations of the product was difficult internally.

Was this for example lower or higher than other project you participated in?

higher in total

What do you think is the cause of this difference?

Due to the smaller group. Therefore everyone needed to take responsibility and everyone needed to work together quickly. This increased the accountability.

Motivation

Interviewer: Compared to previous project you participated in, how was the team's **motivation**?

Very High. higher than other projects

What do you think is the cause of this difference?

Everyone was exited to work in an agile way and to focus on what was most important. And to know that we would see the product a lot faster.

Knowledge

Interviewer: Compared to previous project you participated in, how was the team's **Knowledge sharing**?
Access to the tools was a disadvantage. But ones fixed it was good.
Was this for example lower or higher than other project you participated in? Higher due to a better understanding of how the company worked internally. This made it clearer which information they required and why. This also caused it to provide the information quicker because we knew when they would need it.

Collaboration

Interviewer: Compared to previous project you participated in, how was the team's **collaboration**
It was good.
Was this for example lower or higher than other project you participated in? about the same as other project
What do you think is the cause of this difference?

Product related outcomes

Product quality

Interviewer: Compared to previous project you participated in, What do you think about the **Product quality**?
It is good. about the same as other products we worked on

Interviewer: Compared to previous project you participated in, What do you think about the **Product usability**?
it is also good, meaty higher than other projects
Interviewer: What do you think is the cause of this difference?
We could focus on simplifying the product adding what is most important. So it is quite usable because it is simpler.

Interviewer: Compared to previous project you participated in, What do you think about the **Product bugs / defects**?
It was fine. most of the bugs were related to the generic part of the product (not part of the project)
Interviewer: Is this lower or higher than other project you participated in? A: there were less bugs and defects.
Interviewer: What do you think is the cause of this difference?
because there were less things to implement and few at a time therefore you can do those more correctly/ Interviewer: you have more overview? Yes.

Interviewer: (insurer) Compared to previous project you participated in how **satisfied** are you with the
Very satisfied.
Interviewer: Is this lower or higher than other project you participated in? More satisfied?
Interviewer: What do you think is the cause of this difference?
I think we used our time more efficient and therefore focused on more important things. And working in a way that made a more focused team.

Process related outcomes

Interviewer: Compared to previous projects you participated in how was the **understand ability** of the implementation process
quite clear
Interviewer: Is this lower or higher than other project you participated in? same

Interviewer: Compared to previous projects you participated in how was the **flexibility** of the implementation process
Very high. higher than other projects
Interviewer: What do you think is the cause of this difference?
Due to the open dialogue of what to prioritize and when. That was very flexible. Interviewer: it gave you enough control to make decisions? Yes.

Interviewer: Compared to previous projects you participated in how was the **transparency** of the implementation process
We did not get all the information of what was discussed in the standup meeting. That could have been better.
Interviewer: Is this lower or higher than other project you participated in? It was better.
Interviewer: What do you think is the cause of this difference?
We knew how the company worked internally and how much time it would take and what was difficult for them in the dialogues.

10.8.1.4 *Next steps* ***the company***

Finally I have some questions regarding the application of agile within the company

Interviewer: Would you recommend the company to adopt agile as its main implementation method? Why would you do that? If yes, which parts should be paid additional attention to?
Yes. Because I think it is more efficient. But in other product that begins from the start and people do not have as much experience as us you would need to discuss the process more end to end and set the context of the sprint together. But it was very positive to see the results so quickly. Interviewer: So we would need to spend more time explaining the process and product? Yes.

10.8.2 Interview two

Opening of Interview

Thank you for agreeing to be interviewed for our research on an agile implementation method. We want to find out about your experience with applying agile practices within a EIS implementation. We are collecting information so that we can evaluate the applicability and effects that these practices have. The interview should take about 60 minutes and will be recorded for study later on. Any information that you provide will be anonymized and shall not be used to evaluate your performance as a person in any way. Are you still happy to proceed with the interview?

10.8.2.1 *Demographics*

Let's start with some information that will help me describe your characteristics:

Interviewer: How old are you?

Manager: 35

Note if the respondent is male or female: Male

Interviewer: For which company do you work?

Manager: the company

Interviewer: What is your official job title?

Manager: Manager Senior consultant

Interviewer: What is your role within the Boat project

Manager: Manager and implementer / programmer

Interviewer: Have you any experience with EIS implementations?

Interviewer: how many years?

Manager: 8 year

Interviewer: how many project?

Manager: 15

Interviewer: Have you any experience with agile development methods?

Interviewer: how many years?

Manager : few months

Interviewer: how many project?

Manager: one project

Interviewer: What was your attitude towards an agile implementation method before this pilot?

Manager: very positive on the types of results you might get, but skeptical on the overhead it might Couse

Interviewer: Has this changed in the course of the project?

yes I can see now how you can prevent the overhead, and imp still positive

Interviewer: what has changed?

No

10.8.2.2 *Agile practices*

Next up are some questions about the agile practices that were applied during the project. For each practice I will ask 3 questions. But first, I will provide a short description of the practice in question.

Project charter

The project charter is a high-level summary of the project's key success factors that was created during the exploration meeting on a flipchart-sized sheet of paper.

Interviewer: What was your experience with this practice in the project?
A lot of fun to come together with the team. You sort of get to know each other. It also was a real moment that you feel the projected started. Which I liked. It was also interesting to see how the costumers sees their product. They did not think that much why they sell this product and how many clients use this product. This practice caused them to do that.
Interviewer: Did you face any challenges when applying this practice?
To make the product box was a bit challenging for an insurance product, because the guidelines were created for a new software product you can sell to costumers. Here it was more fixed what you had to build. We did not know what it would look like and how we wild build it. But we pretty much knew this is going to be a boat insurance, and how do you make a product box out of that?
Interviewer: How do you think that this practice affected the outcome of the project?
I am not sure. Somewhere in the back of peoples head this information will be there. But we are not actively doing anything with it Interviewer: so you are quite neutral about it? Well I liked it as a start of a project. But I'm not sure if the information that was wrote down is changing the way we are working now. Interviewer: so it is more of a team building thins? yes

Planning game

The planning game is the practice of determining which user stories should be implemented in the next sprint by assigning priorities to them.

Interviewer: What was your experience with this practice in the project?
It is getting better. In the start it was mostly us that determined how you can split up the product in these steps. And the costumer only indicated things which were less important out of necessity. But now in later sprints you see that it's easier to prioritize. To really say why we are using this and check how many policies are using it and based on that determine the priority. So I think it is an essential part and it can be done pretty well within a configuration. Especially if you have data available with which you can determine priorities.
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
In this case it is always the questions what to do with migrating policies. Because we are making a product that already existed in the old system and are recreating it in the new system with some changes. Then there is always a tendency to make something because it was there in the old system. And I think that will always be a challenge. Interviewer: with striking a balance? Yes. Are they really willing to come up with a new product with priorities meant for selling this product in the market or are they prioritizing based on the old system? Interviewer: is that a challenge faced only by agile project or is that true for other implementations you worked in? I think that will be in different project as well yes.
Interviewer: How do you think that this practice affected the outcome of the project?
This will be very important I think. By looking at priorities every sprint. Then the longer a user story is not picked up the more people realize it is not important. And then people can really say. OK we will not do this, because it has never been important enough. Interviewer: in that sense it will add positively to a project. Yes I think the costumer will become more willing to agree that we will not be doing it. In the old situation they would say we have it now, so we will need it again. And then it is very hard to argue against that. This makes it easier because now they are thinking about why do we actually have that?

Sprint

A sprint is development cycle of fixed length, at the end of which a demonstrable piece of functionality is delivered.

Interviewer: What was your experience with this practice in the project?
So the main benefit that I see from it is that you have to make something that you can demonstrate after the sprint. And because we only have sprint of 2 weeks. You actually have to make something

after 2 weeks.

Interviewer: Is it is a trigger to start working and focus on what is important?

No. it helps / motivates people because you get results and you are not working on paper on some theoretical thing. So people are more motivated because they see results on screen.

Interviewer: Did you face any challenges when applying this practice? And What caused those challenges

There were some difficulties with organizing work in such small periods of time. In terms of planning it was a challenge for everyone. First we tried to make everyone work on the same moment of the week, but because people have other work as well people just worked whenever they had a moment, and some work was done very late in the sprint. Because people were then like: ow wait I need to spend some hours on this. So let's do it last minute. so that was one problem

The other one was when the communication between us and the customer is slow then it is tree days that you cannot work on the user stories. And that has been improving, but still in the last sprint we only got information in the second week on Tuesday.

Interviewing: so it requires some additional agreements and planning?

Yes people really need to make time for it. Multiple moments in the week. So some simple questions we have to ask each other don't have to wait too long.

Interviewer: would a practice like having a common team room would help with that?

Yeah definitely then you don't have to write a massage in a story, but just ask it. Yeah that would definitely help.

Interviewer: How do you think that this practice affected the outcome of the project?

What I said before: it helps with motivations and in the end I think people will enjoy working on this project more. So hopefully they will give you better quality as well. And I think the costumer will be more satisfied, because they had a lot of opportunities to say what they want and come back on their requirements, especial compared to previous project.

Daily meeting

During daily meeting each team members briefly describe any "completed" contributions, any obstacles that stand in their way and their goals for that day

Interviewer: What was your experience with this practice in the project?

I think they were helpful because we did them in the middle of the sprint so it was another trigger for people to get active. It has not helped with collaboration or discussing problems, because we did not do as many of them. Because when you do more you can still keep it short but also give details on specific problems. When there are multiple meetings the chance of someone having a problem he or

she can discuss increases.
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
We did it with audio only and with someone sharing a screen. I think it's better if you see someone's faces. Real life would be best but video is better than no video.
Interviewer: How do you think that this practice affected the outcome of the project?
Yes. It will make it easier to start asking small questions. especially when you are on different locations

Sprint review

At the end of each sprint, a sprint review meeting is held. During this meeting, the Scrum team shows what they accomplished during the sprint. Typically this takes the form of a demo of the new features.

Interviewer: What was your experience with this practice in the project?
Relay good to motivate people. To see the result and quickly sport mistakes or improvements that need to be made. When you see it together as a whole group you can immediately discuss it and find a solutions. Which is much more efficient than waiting until the costumer finds it during testing. Interviewer: is it positive to both the company and the client? Yes. And it was funny to see that we were discussing parts of the generic project, were we had missed something. so that indicates that there is a need for this type of discussions
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
No
Interviewer: How do you think that this practice affected the outcome of the project?
Interviewer: so in summary: motivation. Positive experience. New insights. And discussing thins with a group improves efficiency. Correct.

Task board

Interviewer: What was your experience with this practice in the project?
There were some initial problems to understand it. But for me as a manager its good to quickly get a

overview and see who is working on which ticket. But I would like to have more steps. For example waiting for information
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
New way of working with Jira. Because its digital you only look at it every few days and not daily. It would be better to get a daily reminder. This would be improved by a daily standup in which you look at the board
Interviewer: How do you think that this practice affected the outcome of the project?
It's positive because this gives everyone insights were we are and it's also possible for everyone to manage the status themselves. Therefore the manager has not to do all of those things himself

Sprint Retrospective

Interviewer: What was your experience with this practice in the project?
I was a bid skeptical about it. I thought the comments would be to general. In the end it does work and people came up with improvements. It really helped to get a good feel of what people think is important and how they would like to work in the project.
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
We were not that strict on implementing the improvements. There is not realy much motivation to do it.
Interviewer: How do you think that this practice affected the outcome of the project?
I think it helps people in working together better. Because you learn how and when people like to communicate and what kind of information they need. Compared to a normal project you would hear of those things months later. Or it would go on until it blows up and then it's a big issue. Now because we continuously talk to each other we can solve it in a team and don't need project leaders to do that.

User stories

User stories are requirements that are written on a physical card. These requirements are a functional piece of the product that provides value to the customer and can be implemented independently from each other. This practice helps split the work in small pieces

Interviewer: What was your experience with this practice in the project?
I understand the benefit of writing down the user story so you know who it applies to and what the value is of the story. But for an insurance product it was kind of difficult to find the write the right negative and that was also because we were building something that already exists and not something new. Interviewer: so you mean the formant of "as a" "I want" "because" did not really fit this project? Yes. Interviewer: and what about the splitting the product into smaller parts? That was difficult as well. To make pieces that were independent from each other. Because there are basically tree dimensions in which you can split an insurance product. But you cannot split them across all those dimensions at the same time, because they interact. So that was difficult. However it does make it easier to implement the product because you can do it one at a time and finish it. This makes it simpler and gives you a better overview.
Interviewer: How do you think that this practice affected the outcome of the project?
It helped to divide the work over the team members. Because if we did one big design we could not work on it at the same time. With these smaller pieces we could work on it at the same time. That way we could work quicker and be more dynamic.

Acceptance testing

An acceptance test is a description of the behavior of a software product, expressed as a usage scenario. Executing the acceptance test generally result in either a pass or fail

Interviewer: What was your experience with this practice in the project?
I think the acceptance that we wrote when we were creating the users stories were not detailed enough and it could not really be because we did not have all the information for a specific user story. Even in the sprint planning you do not have the time to spend on acceptance test, because then `you be designing in a session. So the acceptance test that we wrote down did not really help. What we did use were small designs / attachments in Jira and ask question about for example the text that we should use. This was then writing down in the design and implemented. To test you could simply open the design and test based on that instead of on the acceptance test. And I think that works as well as specifying a specific test case. Interviewer: so it does not really help with testing and it does not really add any information to the user story? no.
Interviewer: so that was also the challenge, adding useful details to acceptance tests?

yes
Interviewer: How do you think that this practice affected the outcome of the project?
Not. we did not spend much time on it and it did not really help

Story points

Story points is a measure used to indicate the size of a user story relative to one another

Interviewer: (manager question) How did you experience the estimation of story points?
So first I did it myself for the first sprint and it was quite good to take some time to think about it. And I really liked were you use the number: 1 2 3 5 8, which are a good set of number to choose from. But it was even better when we did it as a team, because then you could get other people's opinion on how much work they thought it was. You can quickly discuss the steps required to implement a story and that helps to come to a quick estimate together. And because the customer hears our discussion they also get a feeling for of how much work it is and why it is so much work and why a story is more work than another story. So it helps their understanding as well. Interviewer: do you think planning poker would help with that? I think maybe with bigger team and the experience level is more equal.
Interviewer: What is your opinion on using story points as an indication of the size of a user story
Yes, it is very useful because we need it to prioritize and see how much we can do in the sprint. Interviewers: and does everyone understand the values? Yes because for the costumer, they really don't know how much work a point is but they can see that one is twice as big as the other.
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
Because of the different levels of experience in the team that determined who was saying anything. So we could not really do it as a team. Interviewer: so this will probably improve in time? Yes and people will learn quicker than they did normally, because we are sizing small things instead of big things.
Interviewer: How do you think that this practice affected the outcome of the project?
I don't know. It's needed as a tool and I think it is positive that it gives the customer some perspective of what is difficult and what is less difficult. Therefor they understand us better and

therefor complain less, because they know why it is more expensive`.

Interviewer: does it affect the prioritization?

Yes it does. We saw today that they wanted a new feature, but when they saw it was a really big user story, they thought well maybe it's better to do the small user stories first that we know we will need. And then wait and see if we can do the new feature in the next sprint.

Velocity

The velocity of a team is the amount of user stories that were completed during that sprint expressed in the units used for effort estimates

Interviewer: What was your experience with this practice in the project?

It was hard to say in the beginning were to begin with and we need more practice to understand it and to get a grip on how much we can do. It's difficult now to say we are doing the right thing.

Interviewer: so it required more data from the past?

Yes and more experience in estimating story points, because that also influences the velocity.

Interviewer: was it user as an indicator of how many points could be implemented in a sprint?

yes we did

Interviewer: and how was the accuracy of that level?

Up to now it was less accurate.

Interviewer: How do you think that this practice affected the outcome of the project?

Neutral or a bid negative, because at one point I think people could have done more. Because when you are so focused on the velocity and people finish their user stories they will not be inclined to do more. It might help if we would have set the velocity `higher than people might have put in more effort to complete those stories.

10.8.2.3 *Project outcomes*

That was the final question about agile practices. Next up are some questions that relate to the projects performance measures. These performance measures are grouped in three categories. First, I will ask some questions about Team related performance attributes. This is followed by some questions about the quality of the product. And I will end with some questions about the implementation process.

Team related outcomes

Productivity

Interviewer: Compared to previous project you participated in, how was the team's **productivity**?

It was higher, so I mean in our hours spend people spend their hours more efficiently I think. I can't prove that, but that's what I think.
What do you think is the cause of this difference?
Because the tasks are smaller and its easer to finish t hem and easier to focus. I think people get less distracted because it is a small thing and people want to finish it, so they get less distracted.

Responsibility

Interviewer: Compared to previous project you participated in, how was the team's sense **responsibility**?
People had problems spending the time they should spend, because there were other products that demanded their attention. And it was not clear what had the higher priority. Maybe that was a project management mistake, so we should have reserved the hours of those people. So that part of the responsibility was not really high. On the other side you could see that people wanted to show up with tickets they completed and show it to the rest of the team. So there was some motivation there.

Motivation

Interviewer: Compared to previous project you participated in, how was the team's **motivation**?
I think that was high and keeps getting higher. Because people enjoy it and want to spend time on it. Because it's a fun project to work on with quick results so I think it increases commitment.

Knowledge

Interviewer: Compared to previous project you participated in, how was the team's **Knowledge sharing**?
It getting better. But that because this is a project in which we work from different locations. And because we now have more contact than that we normally have we are talking about more things in a more efficient way. And also small issues and speak to the whole group instead of one to one. That enables you to share knowledge quickly.
Was this for example lower or higher than other project you participated in? It is less than a project were people are working in the same group. But for a project that is distributed any way I think it is a title better.

Collaboration

Interviewer: Compared to previous project you participated in, how was the team's **collaboration**
That not really better at this moment. When people are working on it it's basically on their own. They meaty share some knowledge.
Was this for example lower or higher than other project you participated in? I think it's less
What do you think is the cause of this difference?
Because it's smaller people can try to come up with something them self and can wait to the last moment to ask for help. In a bigger project you basically need to ask for help otherwise you cannot go further you cannot do anything.

Product related outcomes

Interviewer: Compared to previous project you participated in, What do you think about the **Product quality**?
It's hard to say at this moment.

Interviewer: Compared to previous project you participated in, What do you think about the **Product usability**?
On a few small things you can see we already added a small user story to change the usability aspect. So in that we you can see that we discovered the use ability issues quicker that in a normal project. Interviewer: but the eventual outcome would be the same? no I think this will work better, because sometimes when you are that far in a project and you find a problem in a late stage then it's too expensive and there are other priorities and it will never get changed. And because here you still have sprints left and you have an opportunity to do something about it.

Interviewer: Compared to previous project you participated in, What do you think about the **Product maintainability**? in the start of the project you said that maybe recreating a lot of stuff in the system might add a lot of "trash"
I still have the same opinion. I think we will definitely make some technical design choices at the start that we would not have made if we made a design for the whole project from the start. So that is an issue. It will take a very experienced reviewer to minimize that. It's a new challenge. Interviewer: will this cause problems in the future No I think we have to find a way to work with this, by standardizing and knowing and only using

proven solution we should be able to work like this.

Interviewer: Compared to previous project you participated in, What do you think about the **Product bugs / defects**?

It's the same but we find them quicker.

Interviewer: does that help with solving them?

I think it should help if we can solve them when we are working on the product any way. It could be cheaper, because then you do not need to get back into the project. There is no startup cost, it's still fresh in your memory. Which is not the case when the bug is found after the project is finished. So in that sense it should be cheaper.

Interviewer: Compared to previous project you participated in, What do you think about the **customer satisfaction**?

Much better. Because they really feel that they have something to say and to make small mistake. To them it's really explicate which choices they make. So I think they will be more satisfied with the product in the end and more proud of it.

Process related outcomes

Interviewer: Compared to previous projects you participated in how was the **understand ability** of the implementation process

I think it's clear now after the first few weeks were people have to get used to it. But I think the old process was also very clear, because it only had four steps. So this is a bid more complicated, because you need to have an overview of all tickets and keep track of changing priorities. But I think people will learn quickly and they can work with this.

Interviewer: will this affect the project?

No no, I think in the end there are more benefits to have this process then the negative effect of a bit more complexity.

Interviewer: Compared to previous projects you participated in how was the **flexibility** of the implementation process

It way more flexible, because you can quickly change task from one person to the other if someone is not there. It's easier to put easy tasks with inexperienced employees. So have way more options to process the user stories efficiently. because when its big you

Interview: does it also effect assigning priorities?

Yes because now you continue sly think about requirements and you are allowed to change them even after the third sprint. Whereas in the first situation you have to decide in four weeks and you don't have the chance to change the requirements. or atlas you don't get to see the results before you need to make the decision

Interviewer: Compared to previous projects you participated in how was the **transparency** of the implementation process

I think the improved a lot. Because they understand better what we are doing and why I takes time. so first it was more of a black box to them and now the can be part of the process and we tell them more about technical issues and that makes it more transparent how we work and what our processes are.

10.8.2.4 *Next steps*

Finally I have some questions regarding the application of agile within the company

Interviewer: Would you recommend the company to adopt agile as its main implementation method? Why would you do that? If yes, which parts should be paid additional attention to?

Yes. The biggest challenge is to start up a new project. But once you have the generic deign and some experience with the first product you should definitely be able to do all other product in an agile way and get all the benefits we talked about.

Interviewer: Do you think this method would in larger projects? Why? / Why not? Which parts should be paid additional attention to?

Yes. I'm not really sure yet how to set up multiple products at a time or shorter sprints and one product at a time. But that's something to figure out.

Interviewer: Do you think agile could be used by multiple teams working in parallel?

Why? / Why not? Which parts should be paid additional attention to?

Where there is a good basis then that possible but there needs to be communication between the teams as well otherwise they may have different technical or functional solution.

10.8.3 Interview tree

Opening of Interview

Thank you for agreeing to be interviewed for our research on an agile implementation method. We want to find out about your experience with applying agile practices within a EIS implementation. We are collecting information so that we can evaluate the applicability and effects that these practices have. The interview should take about 60 minutes and will be recorded for study later on. Any information that you provide will be anonymized and shall not be used to evaluate your performance as a person in any way. Are you still happy to proceed with the interview?

10.8.3.1 *Demographics*

Let's start with some information that will help me describe your characteristics:

Interviewer: How old are you?

45

Interviewer: Note if the respondent is male or female: M | F

Female

Interviewer: For which company do you work?

insurer

Interviewer: What is your official job title?

product consultant

Interviewer: What is your role within the Boat project

product developer / lead contact insurer

Interviewer: Have you any experience with EIS implementations?

Interviewer: how many years?

2 year

Interviewer: how many project?

only the current EIS implementation

Interviewer: Have you any experience with agile development methods?

I have studied innovation and development and for that I did a thesis in which I implemented an agile way of working. So I was part of it but this was not my own project. So doing it myself this is my first time I tried it.

Interviewer: how many project?

1 project

Interviewer: What was your attitude towards an agile implementation method before this pilot?

I was very much enthusiastic to try agile

Interviewer: Has this changed in the course of the project?

No, I'm still very positive.

Interviewer: what has changed way and why?

10.8.3.2 *Agile practices*

Next up are some questions about the agile practices that were applied during the project. For each practice I will ask 3 questions. But first, I will provide a short description of the practice in question.

Project charter

The project charter is a high-level summary of the project's key success factors that was created during the exploration meeting on a flipchart-sized sheet of paper.

Interviewer: What was your experience with this practice in the project?
I think it was useful so everyone had the same start and the we were on the same pace and had the same meaning of what we were going to do. And it was also a way of getting to know each other a bid better than before. A good team building experience.
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
We had different backgrounds and some people had no background in agile, whereas I did. So we needed to spend one meeting internally were I needed to repeat how we were going to do things.
Interviewer: How do you think that this practice affected the outcome of the project?
I think it has helped us to be more open and feel more that we do this together and if I compare it how we do it in the other projects I feel like they ask questions and we answer and I don't feel like we build those product together. For this product I feel like we more cooperate.

Interviewer: and why do you think that is the case?

Because we are more open about things if we don't have time or if something is difficult if we don't understand. And when we understand that if we don't help each other then it will be a mutual problem.

Planning game

The planning game is the practice of determining which user stories should be implemented in the next sprint by assigning priorities to them.

Interviewer: What was your experience with this practice in the project?
I depends if you ask about the first or the last sprint. Because in the first one it was a really good help that the manager and you did a lot of work beforehand. And it also helped that we know how to setup product from the ones we done before. Otherwise I think it would be more challenging to prioritize and see what should be in the first sprint. but in the last sprint I think it's easier, because now we have the product in test and it's much easier then to see what u miss and to prioritize. In the first planning games we missed a bid of the overview. Interviewer: would it have helped if there were more user stories in the backlog from the beginning? Yes that would have helped. If we created some big user stories together and detail them later. So that we know we will pick them up in future sprints.
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
For a company that never work agile it's difficult to have everyone understand the same thing. So it took some time and an internal meeting to explain. And to feel like you have the right priority. But having large and more user stories from the start would help with that.
Interviewer: How do you think that this practice affected the outcome of the project?
I think it would be difficult to do without and I think it also helps us to understand that we cannot do everything at the same time and that we need to make priorities.

Product backlog

The product backlog is the list of user stories which at a given moment, are known to be necessary and sufficient to complete a project or a release. In the pilot the product backlog was shown in Jira and held the user stories that still needed to be assigned to a sprint

Interviewer: What was your experience with this practice in the project?
It's a bid split. Because some of the things that are there are less relevant and other things are more important. Interviewer: so it was difficult because some things in the backlog were more important than others? No I'm a bid worried if we can fit all of the important things in the last sprint. Interviewer: but having this overview of the user stories not assigned to a sprint, does that help you? Yes it helps with planning, because it you can see that we discussed and thought about those user stories.
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
no I don't think we had a lot of discussion of what should be in the backlog and what should be in the sprint
Interviewer: How do you think that this practice affected the outcome of the project?

It helps because it gives you the overview you need to have.

Sprint

A sprint is development cycle of fixed length, at the end of which a demonstrable piece of functionality is delivered.

Interviewer: What was your experience with this practice in the project?
I think that the result was very positive. That we can see the results after two weeks helps for the next sprint and for the development itself. It helps us to give more relevant feedback and high quality of the product. But in those two weeks it was challenging to find time and challenging to answer back. Because I might be in tree meetings and cant owner the question for 6 hours. So that's challenging to working in the sprint self to provide information back and forth.
Interviewer: How do you think that this practice affected the outcome of the project?
You are a bid stressed at the end of the two week period instead of the end of the 2 month period. And I think that's positive because then you have more possibility to get things done.

Daily meeting (the company)

During daily meeting each team members briefly describe any "completed" contributions, any obstacles that stand in their way and their goals for that day

Interviewer: What was your experience with this practice in the project?
I think I was the only one from jog attending and it felt part of the team because of it. And I felt like I got some insight information from the developers. What they thought about. I mean we did not know anything about developers before in the other setups. But I'm not sure if there others feel it was necessary to have me there. I'm not sur if my contribution was valuable. Interviewer: but was it value bale to you? Yes. Interviewer: would more members participating from insurer add additional value? No I don't think so.
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
No I think there was no problem.
Interviewer: How do you think that this practice affected the outcome of the project?
I think in necessary to keep up the pace and exchange information. Especial among the developers. Because I could hear they were helping out each other. Interviewer: so it helps with collaboration and coordination? yes

Sprint review

At the end of each sprint, a sprint review meeting is held. During this meeting, the Scrum team shows what they accomplished during the sprint. Typically this takes the form of a demo of the new features.

Interviewer: What was your experience with this practice in the project?
Yeah it's good a necessary. I think we were as much in the test environment as we were in the demo

after the sprint. So a lot of the things that were demonstrated I already knew.

Interviewer: so it did not add much for you?

No.

Interviewer: Did you face any challenges when applying this practice? And What caused those challenges

Perhaps to know me what way we should be prepared.

Interviewer: what information could you like for you to prepare?

I can answer that by saying I would like a list with the things we would need to prepare. Or if we need two more sprints I think everyone would know it as well because it's all about learning.

Interviewer: How do you think that this practice affected the outcome of the project?

I think it helped. I think it has really been a positive experience. I mean we have a product up and running already after just a few weeks. I think me quality for my point of view is much better when I can see it life and then comment on what I see and relate the things that re to come to that instead of seeing a document. Because I have been doing that a lot for the past few years. My experience with the other process is that when they sit and develop after halve a year and they give me question about small details then we have to go back into the document and emails and that takes allot of effort. So that is not required in this way of working so I immediately have an answer.

Task board

The Task board is a board that displays the status of user stories as either: "to do", "in progress", "ready for test" or "done". This board was visible in Jira during the pilot project and showed the user stories that were assigned to a sprint.

Interviewer: What was your experience with this practice in the project?

I perhaps lack some of the details and functionalities of Jira. The more you know the better. And it's also a challenge that we have 4 different Jira setups. That's a bid challenging when you don't work with only one format.

Apart from that I think it's a good way to see the overview and status and it's easy to f in your own tasks and assign them to others. It gives a good overview of your own tasks and sprints and what left in the other sprints.

Interviewer: How do you think that this practice affected the outcome of the project?

I think it give s and better understanding of where the project is and where we need to provide information

Sprint retrospective

The Sprint retrospective is a meeting in which the team reflects on the most significant events that have occurred in that sprint, and take decisions aiming at remediation or improvement

Interviewer: What was your experience with this practice in the project?

I think it was useful. Some things were not so supping. It was also interesting to hear wat the developers felt. Something we discussed in the start was that some information I could have provided without someone asking me because I know that it was important. so that was kind of obvious but I did not think of it myself, so I helped me with being aware of others problems

Interviewer: Did you face any challenges when applying this practice? And What caused those challenges

Sometimes you have some challenges that are not possible to solve.

Interviewer: How do you think that this practice affected the outcome of the project?

That we talk to one another to discover what is important to one another and wat is challenge for the others. And you feel like you are more responsible to feel a solution that is good for the whole team instead of just doing your own small task.

Interviewer: so it helps with team building?
yes

User stories

User stories are requirements that are written on a physical card. These requirements are a functional piece of the product that provides value to the customer and can be implemented independently from each other. This practice helps split the work in small pieces

Interviewer: What was your experience with this practice in the project?
I'm all in favor of user stories, but I need some additional experience to write them correctly. But it's a really smart way to make requirements because it helps you to see / argue why something is relevant to the customer or the company. So it's very good to put things in context and make a good prioritization of the product.
Interviewer: How do you think that this practice affected the outcome of the project?
It helped us to have the costumer force. but we did not really stuck to them, when we put them in Jira it felt like we were back in the old technical way of working, because the user story template were no longer used there Interviewer: so we should be trickster in the format? Yes and we need to have more experience. But I think it will give higher value to the company if we start thinking in user stories and a higher quality for how we spend our time. Interviewers: so it helps prioritize as well? yes

Acceptance testing

An acceptance test is a description of the behavior of a software product, expressed as a usage scenario. Executing the acceptance test generally result in either a pass or fail

Interviewer: What was your experience with this practice in the project?
I think the AT in Jira was ok to check out. But I'm not sure if I quite get the way of writing them the way we should do it. I think I still lack some knowledge. Nut I do feel like I know enough to do the test, but I can't write them myself.
Interviewer: How do you think that this practice affected the outcome of the project?
In reality it was not that much effected. Because we also do it in the old versioned way, because we know we what we want although we are not so good in saying it in a costumer friendly way.

Story points

Story points is a measure used to indicate the size of a user story relative to one another

Interviewer: What is your opinion on using story points as an indication of the size of a user story
I think it useful for us as a costume to see how complex it is, because that is something that was completely hidden for us before. So I think I was interesting to see: ow this is so complicated especially for things I thought were easy. It helps also to build understanding of why we have to prioritize like we do.

Interview: did it help to be present in the discussion of the story point estimates?

Yes it's like Bing in the standup meetings. It's interesting to hear to discuss. But maybe that's because it's new. Maybe when this is the 20th product then it might not provide much new.

Interview: what if you could adjust the scope during the discussion and that way find the best solution given the cost and the benefits.

Yeah that's also relevant that we don't come in and have a fixed idea of what the result should be. But it can help us to get to the conclusion in another way.

Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
None, apart that I can't control that if the stories points that are said are correct.
Interviewer: How do you think that this practice affected the outcome of the project?
I think it helps to see the whole picture and perhaps it gives a good quality because we have to prioritize correctly.

Velocity

The velocity of a team is the amount of user stories that were completed during that sprint expressed in the units used for effort estimates

Interviewer: What was your experience with this practice in the project?
I think it is ok but what has worried me now in the last week is when we find bugs / big problems in what has been developed in the last sprint this will take up a lot in the next sprint and kind of rearranges the priorities and changes the whole picture. That's an uncertainty and it's tempting to say: "they made mistakes and they have to fix it, but if you're on the same team you can't do that. But even when we say you can do 30 points and user that we think are necessary might have to be that there can be 10 of them due to the problems of the previous sprint. Interviewer: and do you think this is new to this project or is it also true for other projects, only then in a later stage? It's definitely not new, but its new that is so visible for us so early. But we have to get used to see it so soon and I think in the total picture it's smart that we are open and see it early. Because when we are on the same team and the goal is to make a product within a certain amount of time. Then you need to find a solution together and maybe not develop something extra.
Interviewer: How do you think that this practice affected the outcome of the project?
I am surprised about how much we finished in each sprint. So I think the time that is calculated for each sprint is enough to complete the whole product.

10.8.3.3 *Project outcomes*

That was the final question about agile practices. Next up are some questions that relate to the projects performance measures. These performance measures are grouped in three categories. First, I will ask some questions about Team related performance attributes. This is followed by some questions about the quality of the product. And I will end with some questions about the implementation process.

Team related outcomes

Productivity

Interviewer: Compared to previous project you participated in, how was the team's **productivity**?

It's too soon to answer to that because since we have so much ales to do at the same time and it has been challenging to find time. And I feel like is should have spent more time on the project than is hold have. Then again we have done a lot in a short time. And what I told you before that it normally takes a lot of time to go back and forth and down details you decided halve a year ago. So that's something we don't have to do.

Interviewer: so the time that was spend was spend efficiently?

yes

Responsibility

Interviewer: Compared to previous project you participated in, how was the team's **responsibility**?
For me personally it's higher because I really believe in this way of working and I want it to be a success. So I kind of feel like we have a good quality and a good process. But one disadvantage is that I don't have the fixed deadlines. Therefore the other project that did have deadlines, those projects got priority. So it's not a good combination with the other way of working.

Motivation

Interviewer: Compared to previous project you participated in, how was the team's **motivation**?
I think everyone was positive and motivated and I also Feld in the other setup that we have become more a team with the company because of boat.
Was this for example lower or higher than other project you participated in? For me this project it is much higher, because I think this is much more fun and this is much more me. I am more the type that like to do a bid of everything and then see results instead of working in sequences. Therefore I'm much more motivated. interviewer: is that also true for the other team members: I think that's mixed, some people are used to the old way of working and like that more. Because you don't have to take that much responsibility and just do what you're asked without thinking for yourself. New things are always scary.
What do you think is the cause of this difference?

Knowledge sharing

Interviewer: Compared to previous project you participated in, how was the team's **Knowledge sharing**?
I think it has been good. I'm a bid reluctant because I think we have been doing well in Theo there project as well. But what we have done here is to include the whole value chain of the product and I think that improved the setup.

Interviewer: therefore it's more efficient? yes
Was this for example lower or higher than other project you participated in?
What do you think is the cause of this difference?

Collaboration

Interviewer: Compared to previous project you participated in, how was the team's **collaboration**
I which I could say yes, but I think time has stopped us to be more collaborative.

Product related outcomes

Quality

Interviewer: Compared to previous project you participated in, What do you think about the **Product quality**?
I'm not sure. It's a bit early to say because we are not finished. And I can just compare it to the document so it's difficult to compare that to a working product. So I could not say right now. But I assume it will be a better quality because we have the possibility to test while developing and it's so helpful instead of sitting months after and se results for the first time. So it would surprise me if this is not better than the others.
Interviewer: Is this lower or higher than other project you participated in?
Interviewer: What do you think is the cause of this difference?

Usability

Interviewer: Compared to previous project you participated in, What do you think about the **Product usability**?
I think that will be improved because it's much easier to see and try the product now. Instead of thinking in a document of this is working or not. And I could bring in a costumer to see the test and get feedback from them for the next sprint.

Bugs / defects

Interviewer: Compared to previous project you participated in, What do you think about the **Product bugs / defects**?
Yeah I'm quite impressed. We have not found that many bugs varying in the other products how many bugs there have been so I'm not sure it's due to agile or due to coincidence.
Interviewer: Is this lower or higher than other project you participated in?
Interviewer: What do you think is the cause of this difference?

I think because the developers know we will test it write away and because it's more interesting or fun of a whole team to set up a project in a limited time.

Satisfied

Interviewer: (insurer) Compared to previous project you participated in how **satisfied** are you with the
I'm ok. It's like we have invented something brilliantly new. This is what we wanted it to be.
Interviewer: Is this lower or higher than other project you participated in? it's about the same

Process related outcomes

Understandability

Interviewer: Compared to previous projects you participated in how was the **understandability** of the implementation process
For me it was understandable.
Interviewer: Is this lower or higher than other project you participated in?
Email received after interview: I took me a long time to understand the waterfall method that … normally uses. I struggled for many months to not get flooded by all of the documents back and forth. And not being able to discuss things in-between workshops. so this was for me the more intuitive way of working I have one extra comment after the interview last week that I think you should incorporate in my feedback. You asked me how easy it was to understand the agile way of working, and I told you that we have had an extra internal meeting to completely understand the process and how to work. First, this was mostly to include other memebers in the project, and to for my sake as the introduction and intial meeting was enough for me to understand the «new» way of working. But morst important: The project members have spent a lot of time to understand the waterfall process. We have had repeatingly meetings and explanations during the whole project and pre-project to be sure everybody understands the documents, the deadlines, how the final product will be, and how it influences the way we work now, and how we will work in the future. If I should compaire the two ways of working from scratch – I you come in as new in a project – the waterfall process is at least 5 times more time consuming and complex to understand. This is because you have a lot of documentation, and extremely long time from you start till you see any result. And my experience is that it always take time to make people understand WHY they shall read and comment documents as this prosess is not intuitive. When we talked in the interview I kind of compaired the agile procedure at the time of start up of this as a new process, with the waterfall process after having practising that for a year. I don't this is fair since the agile way of working is almost easier to understand even with a whole year of experience in waterfall process.

Flexibility

Interviewer: Compared to previous projects you participated in how was the **flexibility** of the implementation process
It is much better it is possible to talk to developers and discuss and see and change around. and only to know that is helpful
Interviewer: Is this lower or higher than other project you participated in?
Interviewer: What do you think is the cause of this difference?
it's the openness and the feeling that we are donor this together and not competitors of different worlds

Transparency

Interviewer: Compared to previous projects you participated in how was the **transparency** of the implementation process
That has one of the bigger positives things because it helps to talk and understand each other. and it's much easier to understand if something is a problem or someone has less time it's easy to understand why things are the way they are.
Interviewer: Is this lower or higher than other project you participated in? In other projects I feel like its necessary to keep your back clean and only show a good facade and not be honest about problems and hope you can fix it. And then suddenly you have a big problem that you did not know about before. if you would have to know it early you might have a suggestion that might help the problem disappear.
Interviewer: What do you think is the cause of this difference?

10.8.3.4 *Next steps the company*

Finally I have some questions regarding the application of agile within the companyv

Interviewer: Would you recommend the company to adopt agile as its main implementation method? Why would you do that? If yes, which parts should be paid additional attention to?
YES, you can write that in big letters. I think it will be a great increase in efficiency and quality.

10.8.4 Interview four

Opening of Interview

Thank you for agreeing to be interviewed for our research on an agile implementation method. We want to find out about your experience with applying agile practices within a EIS implementation. We are collecting information so that we can evaluate the applicability and effects that these practices have. The interview should take about 60 minutes and will be recorded for study later on. Any information that you provide will be anonymized and shall not be used to evaluate your performance as a person in any way. Are you still happy to proceed with the interview?

10.8.4.1 *1.1.1 Demographics*

Let's start with some information that will help me describe your characteristics:

Interviewer: How old are you?
32
Interviewer: Note if the respondent is male or female: M | F

Female
Interviewer: For which company do you work?
keyane
Interviewer: What is your official job title?
consultant
Interviewer: What is your role within the Boat project
developer
Interviewer: Have you any experience with EIS implementations?
3 jaar en 3 maanden
Interviewer: how many years?
Interviewer: how many project?
3 jaar
Interviewer: Have you any experience with agile development methods?
in een ander project waren ze er mee begonnen met een kanban board.
Interviewer: What was your attitude towards an agile implementation method before this pilot?
ik wist het eigenlijk niet helemaal wat ik er van kon verwachten en ik wist niet of het zou werken als je het imidden in een project zou invoeren
Interviewer: Has this changed in the course of the project?
ìk vind het vooral heel leuk. ik weet niet of het invloed heeft op de rest van het project.

10.8.4.2 *Agile practices*

Next up are some questions about the agile practices that were applied during the project. For each practice I will ask 3 questions. But first, I will provide a short description of the practice in question.

Project CHARTER

The project charter is a high-level summary of the project's key success factors that was created during the exploration meeting on a flipchart-sized sheet of paper.

Interviewer: What was your experience with this practice in the project?
het is moeilijk om te bedenken wat er op moet komen te staan en vond dat er niks met de resultaten gedaan werd
Interviewer: How do you think that this practice affected the outcome of the project?
ik denk dat het geen invloed heeft gehad

Planning game

The planning game is the practice of determining which user stories should be implemented in the next sprint by assigning priorities to them.

Interviewer: What was your experience with this practice in the project?
dat je er als de rest van het team als spek en bonen bij zat. interviewer: hoe denk je dat dat komt. omdat de product owner bepaald wat er gaat gebeuren. interview: ook niet om technische input the krijgen?

nee, want dat deed de manager. maar dar heb je niet 5 of 6 mensen voor nodig. bij een ander project deden we ook samen de story points toewijzen en dan ben je er net iets meer bij betrokken dan wanneer de manager al de ponten heeft bepaald. ik had er nu net zo goed niet kunnen zijn.

Interviewer: Did you face any challenges when applying this practice? And What caused those challenges

Interviewer: How do you think that this practice affected the outcome of the project?

zij zijn positief. maar ik denk dat het eigenlijk niet veel invloed heeft.

interviewer: waarom denk je dat dat is?

omdat je eigenlijk al weet wat er moet komen en het past ook wel allemaal moet komen en dat past ook wel allemaal in een sprint ofzo. ik denk dat het project net nog wat t e klein is om echt te zeggen, dit kunnen wel en dit niet doen. wat we nu niet doen zijn eigenlijk allemaal hele kleine dingetjes.

interviewer: dus prioriteiten stellen heeft geen zin in dit soort projecten.

ja, want je weet toch als wat je als eerste gaat doen, omdat anders het product niet gaat werken. dus je weet waar je gaat beginnen.

interviewer: en weet de klant dat ook?

een beetje, omdat ze EIS al wel een beetje kende

Product backlog

The product backlog is the list of user stories which at a given moment, are known to be necessary and sufficient to complete a project or a release. In the pilot the product backlog was shown in Jira and held the user stories that still needed to be assigned to a sprint

Interviewer: What was your experience with this practice in the project?

niet naar gekeken

interviewer: waarom niet?

1e sprint was ik te druk met andere dingen. 2e sprint was ik op vakantie en 3e sprint is nu pas bezig. Voornamelijk de tijds druk, als ik meer tijd had ik er waarschijnlijk wel naar gekeken.

Interviewer: Did you face any challenges when applying this practice? And What caused those challenges

Sprint

A sprint is development cycle of fixed length, at the end of which a demonstrable piece of functionality is delivered.

Interviewer: What was your experience with this practice in the project?

Het is leuk dat je gelijk iets kan zien.

interviewer: en dat je alle stappen zelf doen?

maakt me niet veel uit.

Interviewer: How do you think that this practice affected the outcome of the project?

ik denk dat mensen entausiaster worden.

Daily meeting (the company**)**

During daily meeting each team members briefly describe any "completed" contributions, any obstacles that stand in their way and their goals for that day

Interviewer: What was your experience with this practice in the project?
een keer mee gemaakt en toen was joris aan het praten. Dat was OK, maar of het veel invloed had vraag ik me af.
Interviewer: How do you think that this practice affected the outcome of the project?
Als we het vaker hadden gedaan denk ik het wel. Als je heel de week agile bezig bent en aan hetzelfde project dan is het waarschijnlijk wel nuttig.

Sprint review

At the end of each sprint, a sprint review meeting is held. During this meeting, the Scrum team shows what they accomplished during the sprint. Typically this takes the form of a demo of the new features.

Interviewer: What was your experience with this practice in the project?
dat was goed. dan ziet de klant het en die word enthousiast en helpt om volgende dingen te bepalen en jij word enthousiast. Vooral voor de klant om te bepalen welke tickets daarna moesten gedaan worden.
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
nee. behalve een technisch probleempje met de verschillende omgevingen.

task board

The Task board is a board that displays the status of user stories as either: “to do”, “in progress”, “ready for test” or “done”. This board was visible in Jira during the pilot project and showed the user stories that were assigned to a sprint.

Interviewer: What was your experience with this practice in the project?
ik denk wel positief ik vind het leuk om dingen te slepen en kan zien dat iets af is. het helpt dus met overzicht houden en motivatie
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
nee, was duidelijk

Sprint retrospective

The Sprint retrospective is a meeting in which the team reflects on the most significant events that have occurred in that sprint, and take decisions aiming at remediation or improvement

Interviewer: What was your experience with this practice in the project?
Ik vond het goed, maar we hebben het geen goed vervolg gegeven. interviewer: waar komt dat door? te weinig tijd. interviewer: dus dat was ook de challange van deze practice? ja, je maakt met ze allen zo een lijstje en verziend allemaal verbeteringen, maar dan word er uiteindelijk niks mee gedaan. Dat komt doordat dit een klein onderdeel is van een groter project en daardoor hebben andere onderdelen meer prioriteit.
Interviewer: How do you think that this practice affected the outcome of the project?

we hebben wel goede punten van opgeschreven en mensen worden er op gewezen en dat helpt wel een beetje om iedereen duidelijk te maken wat van ze gevraagd word.

User stories

User stories are requirements that are written on a physical card. These requirements are a functional piece of the product that provides value to the customer and can be implemented independently from each other. This practice helps split the work in small pieces

Interviewer: What was your experience with this practice in the project?
bij het opsplitsen van werk helpt het zeker en je weet precies wat je moet doen en wat er afgerond moet zijn als je klaar bent. interviewer: en is dat handig? ja anders weet je niet wat je moet gaan doen. Maar soms had ik andere onderdelen ook op willen pakken. nu moet ik er de volgende keer weer induiken, dus dat kost meer werk dan wanneer ik het in een keer had gedaan. interviewer: wat vond je van de format zelf? niet zo veel aandacht aan besteed was wel duidelijk.
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
ja dus eigenlijk dat ik wil doorwerken maar dat dat niet kan.
Interviewer: How do you think that this practice affected the outcome of the project?
wel positief dus iedereen weet wat er gaat komen aan het eind van de sprint.

Acceptance testing

An acceptance test is a description of the behavior of a software product, expressed as a usage scenario. Executing the acceptance test generally result in either a pass or fail

Interviewer: What was your experience with this practice in the project?
niet gezien, door een gebrek aan tijd en doordat ik op vakantie was. Daar heb ik me verre van weg geluiden.

Story points

Story points is a measure used to indicate the size of a user story relative to one another

Interviewer: What is your opinion on using story points as an indication of the size of a user story
wel handig maar ik heb het liever gewoon in uren, want dan weet ik niet hoelang ik ergens aan mag besteden. interviewer: en als je zelf weet wat je velocity is. zou dat helpen ja, maar dan heb ik het toch liever in uren. interviewer: zou het helpen om dit samen te doen? dat deden we al een beetje op het einde. maar ik zou het leuk vinden om dat spelletje te spelen. interviewer: zouden de schattingen dan ook beter worden?

ja ik denk het wel want iedereen schat het op zijn eigen niveau. joris zal het toch altijd iets lager inschatten.
Interviewer: How do you think that this practice affected the outcome of the project?
het is wel hadnig voor insurer om te weten hoeveel user stories ze in een sptin kunen plannen. en ze konden zien of iets moeilijker en makkelijker was. dus voor hun word het wel duidelijker. ook om prioriteiten te stellen.

Velocity

The velocity of a team is the amount of user stories that were completed during that sprint expressed in the units used for effort estimates

Interviewer: What was your experience with this practice in the project?
ik denk dat we eindelijk meer story points hadden kunnen doen dan dat we in een sprint deden als iedereen echt 7 uur per week aan hadden besteed. interviewer: dus het was te laag ingeschat? ja.
Interviewer: How do you think that this practice affected the outcome of the project?
meer duidelijkheid naar insurer en wij moeten gewoon weten hoeveel punten we kunnen doen.

10.8.4.3 *1.1.3 Project outcomes*

That was the final question about agile practices. Next up are some questions that relate to the projects performance measures. These performance measures are grouped in tree categories. First, I will ask some questions about Team related performance attributes. This is followed by some questions about the quality of the product. And I will end with some questions about the implementation process.

1.1.3.1 Team related outcomes

productivity

Interviewer: Compared to previous project you participated in, how was the team's **productivity**?
die is wel hoger want iedereen wil zijn dingen afmaken en zijn daarom meer gemotiveerd

responsibility

Interviewer: Compared to previous project you participated in, how was the team's **responsibility**?
dat verschilt joris voelt zich het verantwoordelijkst en hij wil dat alles af is. en de rest is variërend.
Was his for example lower or higher than other project you participated in ? maakt niet uit.

motivation

Interviewer: Compared to previous project you participated in, how was the team's **motivation**?
die was bij sommige wat hoger en andere lager. en in het einde was het hoger dan in het begin.
Was this for example lower or higher than other project you participated in ? ik denk gemotiveerder
What do you think is the cause of this difference?
omdat het nieuw is en je snel resultaten ziet.

Knowledge

Interviewer: Compared to previous project you participated in, how was the team's **Knowledge sharing**?

niet anders dat normaal
Was this for example lower or higher than other project you participated in ? het was meer dat iedereen deed wat hij kon oppakken. dus daarom was je niet echt met nieuwe dingen bezig en daarom hoefde je ook niks te vragen. interviewer: en het vragen stellen aan insurer? dat ging op het begin wat moeilijk. er werd niet helemaal begrepen wat de vraag was. soms denk ik dat ze gewoon niet genoeg moeite in stopte om de vragen te snappen. Maar dat schijnt verbeterd te zijn. interviewer en als je dat vergelijkt met andere project, verschilt dat dan? ne eigenlijk niet.
What do you think is the cause of this difference?

collaboration

Interviewer: Compared to previous project you participated in, how was the team's **collaboration**
iedereen was met zijn eigen dingen bezig.
Was this for example lower or higher than other project you participated in ? nee maar dan werk je aan een groter ding wat voor jou is. dus eigenlijk is er geen verschil

1.1.3.2 Product related outcomes

quality

Interviewer: Compared to previous project you participated in, What do you think about the **Product quality**?
ik denk dat het beter is omdat je het gelijk ziet iedere keer. daardoor kan je het verbeteren. wanneer je het in een keer doet dan heb je die mogelijkheid niet.

usability

Interviewer: Compared to previous project you participated in, What do you think about the **Product usability**?
niet veel verschil, uitenlijk word het wel bijgewerkt. nu kun je het alleen eerder zien.

maintainability

Interviewer: Compared to previous project you participated in, What do you think about the **Product maintainability**?
ik denk niet dat het verschil maakt.

bugs / defects

Interviewer: Compared to previous project you participated in, What do you think about the **Product bugs / defects**?
die zijn kleiner omdat het allemaal kleine stukje zijn en je aan jou stukje werkt kun je dat beter testen dan wanneer je een groot ding test. wanneer er dan iets niet klopt dan los je dat weer op. dus ik denk dat er minder bugs zijn.

satisfaction

Interviewer: Compared to previous project you participated in, What do you think about the **customer satisfaction**?
die is hoger. omdat ze gelijk dingen kunnen zien.

1.1.3.3 process related outcomes

understandability

Interviewer: Compared to previous projects you participated in how was the **understandability** of the implementation process
niet helemaal begrijpbaar, omdat ik niet helemaal weet welke namen waar bij horen. maar dat komt door de hoeveelheid tijd die ik er in heb gestoken. ik vraag me wel af hoe het is als alle projecten zo gaan doen. of het dan nog wel overzichtelijk is.

flexibility

Interviewer: Compared to previous projects you participated in how was the **flexibility** of the implementation process
niet zo veel verschil. zolang er niet aan iets gewerkt word kan er via een change request nog best iets veranderen en dat zie ik in beide manieren van werken terug komen.

transparency

Interviewer: Compared to previous projects you participated in how was the **transparency** of the implementation process
er waren geen geheimen, dus er werd gewoon verteld als er iets niet zo goed ging.
Interviewer: Is this lower or higher than other project you participated in ? ik vind dat je altijd eerlijk meot zijn, maar ik denk dat je in dit project meer open bent.
Interviewer: What do you think is the cause of this difference?
wel goed voor de relatie naar de klant want ze vertrouwen je wel.

10.8.4.4 *1.1.4 Next steps the company*

Finally I have some questions regarding the application of agile within the company

recommendation

Interviewer: Would you recommend the company to adopt agile as its main implementation method? Why would you do that? If yes, which parts should be paid additional attention to?
ja maar ik ben huiverig over als het groter word.
Interviewer: Do you think this method would in larger projects? Why? / why not? Which parts should be paid additional attention to?
dat het overzicht weg is. dat de lijst van dingen die je moet oppakken te groot is. en ik dnek dat je wel moet blijven focusen op een product en niet andere producten er naast en dan snel opleveren. anders raken mensen de kluts kwijt en weten ze niet met welk prodccut ze bezig zijn.
Interviewer: Do you think agile could be used by multiple teams working in parallel? Why? / why not? Which parts should be paid additional attention to? ja, maar het is wel lastig dat het ene team iets technisch word bepaald en dan door het andere team ook gebruikt moeten worden. interviewer: dus er moet goede communicatie plaats vinden.

10.8.5 Interview five

Opening of Interview

Thank you for agreeing to be interviewed for our research on an agile implementation method. We want to find out about your experience with applying agile practices within an EIS implementation. We are collecting information so that we can evaluate the applicability and effects that these

practices have. The interview should take about 60 minutes and will be recorded for study later on. Any information that you provide will be anonymized and shall not be used to evaluate your performance as a person in any way. Are you still happy to proceed with the interview?

10.8.5.1 *Demographics*

Let's start with some information that will help me describe your characteristics:

Interviewer: How old are you?
25
Interviewer: Note if the respondent is male or female: M | F
Female
Interviewer: For which company do you work?
the company
Interviewer: What is your official job title?
junior consultant
Interviewer: What is your role within the Boat project
developer
Interviewer: Have you any experience with EIS implementations?
Interviewer: how many years?
1 year and a few months
Interviewer: how many project?
3
Interviewer: Have you any experience with agile development methods?
No
Interviewer: how many years?
Interviewer: how many project?
Interviewer: What was your attitude towards an agile implementation method before this pilot?
Good. it was positive
Interviewer: Has this changed in the course of the project?
no

10.8.5.2 *Agile practices*

Next up are some questions about the agile practices that were applied during the project. For each practice I will ask 3 questions. But first, I will provide a short description of the practice in question.

Project CHARTER

The project charter is a high-level summary of the project's key success factors that was created during the exploration meeting on a flipchart-sized sheet of paper.

Interviewer: What was your experience with this practice in the project?
Good to get a shared view of how we are working and what we expect this for be successful.

Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
Yeah difficult to get something useful and understandable for the team and know what you are looking for as a team. Interviewer: is that because that this practice is new to you? Yeah. You know what is supposed to happen to get a good product. But for the scrum way of working what should be a good way is unclear. So it's a learning process.
Interviewer: How do you think that this practice affected the outcome of the project?
It was a good way of starting the project and get everyone's mindset in the scrum way of working and knowing that we are doing something different for this project. However it is not used much later, because we set those criteria and then did not look at them later. Then we just slip back in our old way of working.

Planning game

The planning game is the practice of determining which user stories should be implemented in the next sprint by assigning priorities to them.

Interviewer: What was your experience with this practice in the project?
Good for them to think about what they needed and what was more important than other things. Because in other project they used a lot of time on non-important stuff. So here they needed to say them self what was less important and then needed to follow that through. Interviewer: and did it help for you as a developer? The thing that were on the board I worked on and did not look so much on the backlog. For example the prioritization and sizing of the stories I did not spend much time on them. I just did what was included in the sprint. Interviewer: so you just focused on year tasks? yes
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
No I think it was... in the beginning it was hard to prioritize. But afterwards it gets better. So I think experience is important. So if this was a longer project then I would improve.
Interviewer: How do you think that this practice affected the outcome of the project?
Sure it was a good way of sizing things up and including them in the sprint. And we only could handed a certain amount so if things were full they really needed to prioritize. Now we have 35 points in the sprints and that could be increased as well because our working could be more efficient. Interviewer: so it helps the project to prioritize even more than other project you have been in? Yes.

Product backlog

The product backlog is the list of user stories which at a given moment, are known to be necessary and sufficient to complete a project or a release. In the pilot the product backlog was shown in Jira and held the user stories that still needed to be assigned to a sprint

Interviewer: What was your experience with this practice in the project?
It's good to know what is coming and what is planned to be implement in the future. And I can see if something is missing in another user story is can see if tis include in a different user story on the

backlog. Just know that it is evolving and work with it as well.
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
In the beginning there was not that much on the backlog. That made it difficult to see the whole product r the whole project. Interviewer: so if the backlog was filled from the start that would have helped? Yes. And we did not follow the format throughout the whole project, but for me that does not matter.
Interviewer: How do you think that this practice affected the outcome of the project?

Sprint

A sprint is development cycle of fixed length, at the end of which a demonstrable piece of functionality is delivered.

Interviewer: What was your experience with this practice in the project?
I think the length was good but we were so under presure for other projects that it could not be prioritized. We started out to say we would work in the same timeslots, but that was not really possible because you sometimes required information to continue, so you had to wait. Later we agreed to respond to mails and questions within a certain amount of time, which that gave more flexibility. Currently I just check my user stories and ask for the information I need at the start of the sprint. Then when I want to start implemented I have everything I need.
Interviewer: How do you think that this practice affected the outcome of the project?
I think it is good to both us and insurer to see fast results and see the system evolving. And that you do all of the steps of doing a design to implementing it. So it's fun to complete something and not just do 10% and then hand it over to someone else.

Daily meeting (the company**)**

During daily meeting each team members briefly describe any "completed" contributions, any obstacles that stand in their way and their goals for that day

Interviewer: What was your experience with this practice in the project?
Maybe unnecessary, because we only had so few. So we were only picking on what we should work on and we could ask questions online. So in this project I think it did not add much. Interviewer: would more meetings have helped? If the project was larger I think it would be good to have this meeting, but because this is only a small product I don't think it's useful, because we can have contact via mail.
Interviewer: How do you think that this practice affected the outcome of the project?
No effect.

Sprint review

At the end of each sprint, a sprint review meeting is held. During this meeting, the Scrum team shows what they accomplished during the sprint. Typically this takes the form of a demo of the new features.

Interviewer: What was your experience with this practice in the project?
good it's fun to show the result so far and see that the client is happy and that the product is evolving
Interviewer: Did you face any challenges when applying this practice? And What caused those

challenges
no
Interviewer: How do you think that this practice affected the outcome of the project?
God. The client was very happy and they could see parts that they had never seen before. Interviewer: so it helps with motivation, anything else? Yeah they solved some bugs and other things they wanted to improve.

Task board

The Task board is a board that displays the status of user stories as either: “to do”, “in progress”, “ready for test” or “done”. This board was visible in Jira during the pilot project and showed the user stories that were assigned to a sprint.

Interviewer: What was your experience with this practice in the project?
It was a good tool to get an overview of the sprint to know what’s still up for grabs and what is finished. It would however be nice if something was completed to see who had worked on it so you would know who to contact about that user story. I also was not put on watching automatically so I did not know when someone replied.
Interviewer: How do you think that this practice affected the outcome of the project?
It helped to get and overview and how things are evolving and how far along we are for a sprint. In the beginning you see 20 thins to do and during you can see that change.

Sprint retrospective

The Sprint retrospective is a meeting in which the team reflects on the most significant events that have occurred in that sprint, and take decisions aiming at remediation or improvement

Interviewer: What was your experience with this practice in the project?
Both positive and negative. It used a lot of time for weird stuff people had. But it helped with coming up with improvements. But we need to do something in those meetings as well.
Interviewer: Did you face any challenges when applying this practice? And What caused those challenges
It took a lot of time that we could have spent on developing the product.
Interviewer: How do you think that this practice affected the outcome of the project?
So it was positive, but it could be done more efficient or another way then just writing post it notes.

User stories

User stories are requirements that are written on a physical card. These requirements are a functional piece of the product that provides value to the customer and can be implemented independently from each other. This practice helps split the work in small pieces

Interviewer: What was your experience with this practice in the project?
It was good. It was good to follow up from the start to the beginning and see something the whole

way trough. To have a notion of how you want to do it and then also do it. There were some dependencies between user stories and then you get a lot of extra work, because some things needed to be done before you could start your task. So maybe we as developers could have directed a bit more during the selection of user stories to make sure there was a good sequence in what was implemented when.

Interviewer: and what did you think about the format?

in the beginning we did it as a disruption of why a customer wants something, but halve way we switched to a description of what was needed and how that could be realized. However insurer said that they preferred the other way. But for me an s a developer I don't mind how it is written down as long as all the information is there.

Interviewer: How do you think that this practice affected the outcome of the project?

It's easier to split up the work between team members and especially when we work from different locations it's important that we don't depend on each other's work.

Acceptance testing

An acceptance test is a description of the behavior of a software product, expressed as a usage scenario. Executing the acceptance test generally result in either a pass or fail

Interviewer: What was your experience with this practice in the project?

I did not look at it, but I did not think it would provide information that would help me.

Story points

Story points is a measure used to indicate the size of a user story relative to one another

Interviewer: (manager question) How did you experience the estimation of story points?

I think it was nice to know what juris thought about the task. But during the task I did not think much about it. Somewhere to small some too big

Interviewer: do you think that estimating it together would have helped?

no but it hank it was good to say this is a stain amount of points and the total amount cannot be more than this and that I as a developer should pick up at least so much.

Interviewer: so it helped select the amount of tasks you should do.

Yes I know when I finished 15 points that I could stop and that some other member should also do their part although I had some additional time.

Interviewer: in another project you would have continued working?

Yeah if this was a fulltime project I would have kept on working.

Interviewer: so this stops you to work?

Yeah because other people require the experience so I won't pick up their work. And one week you work a bid more and the other a bid less depending on how well the size was estimated. I know that we would finish before the demo. But I would have kept on working if I knew we should have prioritized it.

Interviewer: How do you think that this practice affected the outcome of the project?

Not that much. When I set on a thing to be done I just do it. If I knew no one else would pick it up I would just do it. I think it is good to set a limit to the amount you can do as well.

Interviewer: so you can combine multiple projects?

Yeah so you can do it when you have time and don't have to work on set hours.

Velocity

The velocity of a team is the amount of user stories that were completed during that sprint expressed in the units used for effort estimates

Interviewer: What was your experience with this practice in the project?

It's good to say we have an upper limit of what we can do in a sprint. But I think overtime we will know better the amount of points that we can do. So now we have set 35, but I don't think anyone is putting in the hours they should do. I think we could have done twice as much. But although we are not working all of their hours we are still completing all of the product.

Interviewer: Did you face any challenges when applying this practice? And What caused those challenges

It is a learning experience to set the right amount to set for a sprint. A know how much time actually people have and how many points they can do in that time period. But for this short project it work fine.

10.8.5.3 *Project outcomes*

That was the final question about agile practices. Next up are some questions that relate to the projects performance measures. These performance measures are grouped in three categories. First, I will ask some questions about Team related performance attributes. This is followed by some questions about the quality of the product. And I will end with some questions about the implementation process.

Team related outcomes

Productivity

Interviewer: Compared to previous project you participated in, how was the team's **productivity**?
It was high during certain hours but extremely low during other hours. We worked a lot in the end of the sprint.
Was this for example lower or higher than other project you participated in? This is the same in other projects you always need to prioritize the next deadline. You just have to set time and work on that project during that time. Interviewer: so you completed the same amount of work in the same amount of time in this project as other projects? maybe more, because when you start reading the user stories and you get an idea of how to do it you don't have to hand it over so that makes it quicker.

Responsibility

Interviewer: Compared to previous project you participated in, how was the team's **responsibility**?
There is always someone that is more responsible than the rest.
Was this for example lower or higher than other project you participated in? This is always the case.

Motivation

Interviewer: Compared to previous project you participated in, how was the team's **motivation**?
We are all motivated to get everything completed.
Was this for example lower or higher than other project you participated in? It is the same.

Knowledge

Interviewer: Compared to previous project you participated in, how was the team's **Knowledge sharing**?
It's the same as well.

Collaboration

Interviewer: Compared to previous project you participated in, how was the team's **collaboration**
Better because there is more contact or if there is something you are missing or wondering about. Then you cannot wait for a long time to complete it.

Product related outcomes

Quality

Interviewer: Compared to previous project you participated in, What do you think about the **Product quality**?
It's good. But I don't think we would have come up with the same solution if we did it in the normal way.
Interviewer: Is this lower or higher than other project you participated in? Maybe better because we have more control over it. In this project we don't spend a lot of time on weird / detailed stuff. Therefore it's cleaner and you don't get distracted by all those things. Those things are pushed to the end because they are less important. Those normally take a lot of time.

Usability

Interviewer: Compared to previous project you participated in, What do you think about the **Product usability**?
I think it's good.
Interviewer: Is this lower or higher than other project you participated in? No I would not say so.

Maintainability

Interviewer: Compared to previous project you participated in, What do you think about the **Product maintainability**?
No I think that maintaining all the documents could be more difficult because different people work on it that have different styles. That might be confusing and effect the total overview. But other than that I don't think this is an issue.

Bugs / defects

Interviewer: Compared to previous project you participated in, What do you think about the **Product bugs / defects**?
It's low.
Interviewer: Is this lower or higher than other project you participated in? lower than in other projects
Interviewer: What do you think is the cause of this difference?
I think because of the user stories and you want to following something trough. Therefore you know what should be delivered and it's not a big package. There you don't have the total overview of all the functionality. So when you develop a user story you know it should work like this and it's connected to that part. Therefor you can easily check if it's implemented correctly.

Satisfaction

Interviewer: Compared to previous project you participated in, What do you think about the **customer satisfaction**?
I think it has increased. I think they are much happier to see this product than other. Because they have seen result so fast.

Process related outcomes

Understandability

Interviewer: Compared to previous projects you participated in how was the **understandability** of the implementation process
Yeah I asked some questions to be sure what all the steps were that needed to be done. But other than that it was understandable.
Interviewer: Is this lower or higher than other project you participated in? Maybe better.
Interviewer: What do you think is the cause of this difference?
It's easier to get a shared point of view for things to be done. You can focus on a small picture other than a bigger one when you have the whole product. So it's easier to get a clear answer from insurer

as well.

Flexibility

Interviewer: Compared to previous projects you participated in how was the **flexibility** of the implementation process
It was good. The two weeks sprint were good and that you should deliver in that much time. And know what should be done and that I can select my own user stories and prioritize them myself. And I could do them in my own way. So that gave me more flexibility. For example I could start with to boring ones and end with the fun ones.

Transparency

Interviewer: Compared to previous projects you participated in how was the **transparency** of the implementation process
Yeah I think we are more open about things that are difficult or that we are struggling with and if we can't deliver something we are honest to insurer about why. So they get an insight of how we are working.

10.8.5.4 *Next steps* ***the company***

Finally I have some questions regarding the application of agile within the company

Recommendation

Interviewer: Would you recommend the company to adopt agile as its main implementation method? Why would you do that? If yes, which parts should be paid additional attention to?
Yeah. I think so. But maybe make it more fulltime because I'm not sure if it's a good solution with other work as well. Because now we mixed it with other work and products ad projects.
Interviewer: what would be your main point for recommending it?
It's nice to follow through on things you take responsibility for and you see results and the client is happy. And the client can also prioritize work and get a better view of why something is important for their product.

10.9 Appendix I: interview coding

question	interview one	interview two	interview tree	interview four	interview five
Demographics					
How old are you?	32	35	45	32	25
sex	Female	male	Female	Female	Female
employer	transcendent group	the company	insurer	the company	the company
job title	senior consultant	senior consultant	Product consultant	consultant	junior consultant
role	representative insurer	manager and developer	product owner / customer	developer	developer
year of experience	7	8	2	3	1
number of projects		15	1	3	3
years of agile experiance	0	< 1 year	< 1 year	0	0
number of agile projects	0	1	1	<1	0
attitude	4	5	5	3	4
change in attitude	no change	no change	no change	slightly more positive	no change

question	interview one	interview two	interview tree	interview four	interview five	average
Agile practicces						
Project charter						
experinace	4	5	4	2	4	4
challanges	1	2	2	2	3	2
outcome	4	4	4	3	3	4
Planning game						
experinace	5	3	3	2	4	3
challanges	4	4	3	n/a	5	4
outcome	5	5	4	3	4	4
Product backlog						
experinace	4	n/a	3	n/a	4	4
challanges	4	n/a	5	n/a	3	4
outcome	5	n/a	4	n/a	n/a	5
sprint						
experinace	4	4	5	4	2	4
challanges	3	3	3	n/a	3	3
outcome	4	4	4	4	4	4
daily meeting						
experinace	n/a	4	4	2	2	3
challanges	n/a	3	5	n/a	n/a	4
outcome	n/a	4	4	3	2	3
sprint review						
experinace	5	5	3	4	5	4
challanges	4	5	4	4	5	4
outcome	5	5	4	n/a	4	5
task board						
experinace	4	4	4	4	4	4
challanges	4	4	n/a	5	n/a	4
outcome	4	4	4	n/a	4	4
retrospective						
experinace	3	4	4	3	3	3
challanges	3	3	4	2	3	3
outcome	4	5	4	4	4	4
User stories						
experinace	3	2	4	5	4	4
challanges	2	2	4	3	3	3
outcome	3	4	3	4	4	4

acceptance testing						
experinace	3	2	2	n/a	3	3
challanges	4	2	3	n/a	n/a	3
outcome	3	3	3	n/a	2	3
Story points						
experinace	5	5	4	2	3	4
challanges	4	4	4	4	n/a	4
outcome	4	4	4	4	3	4
Velocity						
experinace	4	3	3	3	4	3
challanges	n/a	2	n/a	n/a	4	3
outcome	4	2	4	3	n/a	3
total						
experinace	4	4	4	3	4	4
challanges	3	3	4	3	4	3
outcome	4	4	4	4	3	4

question	interview one	interview two	interview tree	intreview four	interview five	average
Projet outcomes						
team related outcomes						
productivity	4	4	N/A	4	3	4
sense of responsibility		4	4	3	3	4
motivation		5	4	4	3	4
knowledge sharing		4	3	3	3	3
colabiration		3	2	3	4	3
Product related outcomes						
product quality	3	n/a	n/a	4	3	3
product usability	4	4	4	3	3	4
products mainainability	n/a	2	n/a	3	2	2
product defects / bugs	2	3	2	2	2	2
customer satisfaction	4	5	3	4	4	4
Process related outcomes						
understandability	3	2	4	2	3	3
flexibility	5	4	5	3	4	4
transparency	4	5	5	4	4	4
total	4	4	4	3	3	3

10.10 Appendix J: survey invitation

Hi everyone,

I would like to ask you to participate in a short survey, which should take about 15 minutes to complete.

With this survey I can assess the agility level of our pilot project and compare it with industry benchmarks.

Any information that you provide will be anonymized and encrypted and will not be used to evaluate you as a person.

You can start the survey by following this link: https://goo.gl/eTxFFt

Thank you for participating.

Kind regards,

Jochem van Eldijk

10.11 Appendix K: recommendation to the company

Based on the results of this report I would recommend the company to continue its use of the agile implementation method. However I would recommend the company to have one additional evaluation of the pilot project once it is completed. This evaluation should focus on the quality and maintainability of the product, its documentation and the total amount of time that was spend on its development.

Together with the results of this report the company should decide if they which to continue with the use of the agile implementation method. If after this evaluation the company decides its wants to continue its use of the method I would recommend the follow:

- Use the power point presentations that were made for the pilot as a template for new projects.
- Use the pilot project as an example to explain the way of working
- Spend more time on explaining how to create user stories to the client. For example by having a workshop in which they could practice.
- Do not set a velocity limit in the first sprint. Simply try to implement as many user stories within the available time. In the second sprint the amount of story points that were completed can be used as an initial value of velocity. Keep updating this value as the project progresses. Also keep track the velocity of individual project members. This way you will know how the projects velocity will be affected if someone is unable to work for a certain period of time.
- Try to closely involve the product owner in the projects. If possible have him / her available on premise.
- Create a complete list of user stories before starting the project. This will help the client with prioritizing and getting an overview of work to come. Only those user stories that will be worked on in the near future require a higher level of detail. Others can be left intentionally vague; this way you don't spend too much time on requirements that might still change in the future.
- Further optimize Jira for the use of agile. For example by enabling the user story, task and epic issue types. Or by automatically putting people on watch if they created or worked on a user story, this way they won't miss any new information.

Then there are some points that require additional attention / research:

1. Greenfield project that have no generic part set up.
2. How to distribute work amongst employees with different experience levels and different salaries.
3. How to continuously review functional and technical designs without blocking the projects progress.
4. How to let different teams work in parallel within the same project.
5. How to improve on the quick and sometimes inefficient design decisions that are made in a sprint.
6. How to prioritize and distribute work when employees work on multiple projects.

1. Greenfield project

Personally I can come up with two ways to handle the first attention point. One option would be to either use the old implementation method to come up with a generic design. The other option would be to use the agile method a start with making a first product, based on that product generic parts could be identified.

2. Work distribution

On possible solution to the second point could be letting more experienced employees do the requirement analyses, functional design and technical design and to let the less experienced employees do the configuration and testing. in this situation some additional time would probably need to be spent on handing over the requirements.

3. Reviewing work

The third point could be handled by having at least two people on a implementation team that can review the others designs. However, for small projects this would be a costly solution and because the members would not be outside of the project this could damage the quality of the reviews. Another solution would therefore be to have a group of reviewers that are available on a short notice. Testing both options in practice should show which option results in the best quality of reviews and least amount of time spend waiting on reviews.

4. Parallel teams

In case multiple teams work on different products for the same projects some additional coordination will be required to make sure the teams do not need to invent the wheel twice. An agile way of organizing this coordination is called the scrum of scrums. The agile alliance provides the following explanation of the concept:

“A technique to scale Scrum up to large groups (over a dozen people), consisting of dividing the groups into Agile teams of 5-10. Each daily scrum within a sub-team ends by designating one member as "ambassador" to participate in a daily meeting with ambassadors from other teams, called the Scrum of Scrums.

Depending on the context, ambassadors may be technical contributors, or each team's Scrum Master, or even managers of each team.

The Scrum of Scrums proceeds otherwise as a normal daily meeting, with ambassadors reporting completions, next steps and impediments on behalf of the teams they represent. Resolution of impediments is expected to focus on the challenges of coordination between the teams; solutions may entail agreeing to interfaces between teams, negotiating responsibility boundaries, etc.

The Scrum of Scrum will track these items via a backlog of its own, where each item contributes to improving between-team coordination.” (Agile Alliance, 2015)

5. Refactoring

As the product evolves it might become apparent that some design decisions that were made in the past would not be made with the current knowledge. One way to deal with this issue is to “refactor” parts of product that are already implemented. This agile practice states that changes ae made to the system that it does not alter its external behavior, yet improves the internal structure. A important part of this practice is convincing the client that it’s important to spend some time on refactoring in order to guarantee the systems maintainability and efficiency.

6. Multiple projects

In case employees need two work on multiple products, a managers needs to decide on the priority of the two projects and determine the amount of time that should be spend on those projects accordingly. After one sprint this distribution will result in a velocity for each of the projects. In case priories change multiple steps can be taken:

a. Reduce the velocity of one of the projects and increase the other.
b. Readjust the amount of time that the team can spend on each project and determine a new velocity level was done in the start of the project.

In both cases the manager will need to discuss the implication with both clients and come to a mutual agreement beforehand.