



Expertise
and insight
for the future

Essi Sjöblom

Proposing Improvements for a System Development Process

Metropolia University of Applied Sciences

Master of Engineering

Business Informatics

Master's Thesis

20 November 2020

Author Title	Essi Sjöblom Proposing Improvements for a System Development Process
Number of Pages Date	79 pages + 4 appendices 20 November 2020
Degree	Master of Engineering
Degree Programme	Business Informatics
Instructor	Zinaida Grabovskaia, PhL, Senior Lecturer [REDACTED]
<p>The case company of this thesis is a provider of financial process automation solutions, both customized and standardized. The Product Development Operations are vital for the case company's business to address the customers' needs and, most importantly, to support the sales to have continually working system with updated information security measures and with cutting edge technology and features to boost the sales.</p> <p>This thesis was triggered by the business challenge of bottlenecks and hindrances in the Development Process of a Standardized P2P system. The objective of this thesis was to propose improvements to the Development Process of a Standardized P2P system.</p> <p>The thesis was conducted using qualitative action research methodology. The primary methods of data collection was interviews. Other data methods used were internal documentation analysis, co-creation in meetings and workshops, and observations. The current state analysis resulted in mapping the process and identified the development needs in the Product Development Operations. The existing literature and best practice were selected to reflect the focus areas of the identified development needs grouped into three categories; Process, Communication and Requirements. As an outcome of the literature search, the conceptual framework was built that selected the tools for proposal building.</p> <p>Based on the results of the current state analysis and the conceptual framework, the initial proposal was built with a project team. The initial proposal was validated by three supervisors and based on the feedback, the Final Proposal of Improvements to the Development Process of the Standardized P2P system was developed. The validation also included the discussion about the ways of its implementation. Additionally, it was acknowledged that in agile environment, the process improvements are being sought continuously.</p> <p>By implementing the improvements of the final proposal, the identified development needs related to communication, requirements and process are addressed. Based on the feedback, the agile nature of the improvements elevated the maturity of the process agility, and improved the process structure especially by applying the Scrum framework related improvements.</p>	
Keywords	Product development, system development, process development, P2P, Procurement-to-Pay, Agile, Scrum

Contents

List of Figures

List of Tables

1	Introduction	8
1.1	Case Company and Business Context	8
1.2	Business Challenge, Objective and Outcome	9
1.3	Thesis Outline	10
2	Method and Material	11
2.1	Research Approach	11
2.2	Research Design	12
2.3	Data Collection and Analysis	14
2.4	Research Quality Criteria for This Thesis	17
3	Current State Analysis of the Product Development Operations and the Development Process of the Standardized P2P System	19
3.1	Overview of the Current State Analysis	19
3.2	Description and Analysis of the Current Product Development Operations	19
3.2.1	The Current Product Development Process and Teams	20
3.2.2	Description of the Current Sprints and Ticket Lifecycles	23
3.3	Development Needs of the Current Product Development Process	27
3.4	Description and Analysis of the Development Operations after Corporate Changes	29
3.4.1	Descriptions of the Development Process Operations and Teams after Corporate Changes	29
3.4.2	Descriptions of the Sprints and Ticket Lifecycles after Corporate Changes	30
3.5	Analysis and Key Findings of the Current State of the Product Development Process	33
3.5.1	Analysis of the Changes Made to the Product Development Process	34
3.5.2	Analysis of the Development Needs to the Product Development Process after Corporate Changes	35
3.5.3	Analysis and Comparison of the Development Needs to the Product Development Process before and after Corporate Changes	36
3.5.4	Selected Focus Areas of the Development Needs of the Product Development Process	39
4	Existing Knowledge and Selected Best Practice on Improving Agile Processes	40

4.1	Agile Process Structure and Definition	40
4.1.1	Agile Manifesto and Agile Principles	40
4.1.2	Process Definitions in Agile	42
4.1.3	Scrum	42
4.2	Process Development for Agile	47
4.2.1	Agile Fluency™ Model	47
4.2.2	Agile Process Maturity	50
4.3	Communication in an Agile Process	51
4.4	Requirements and Specifications in an Agile Process	52
4.5	Conceptual Framework of This Thesis	54
5	Building Proposal for Improvements to the Development Process of a Standardized P2P System	57
5.1	Overview of the Proposal Building Stage	57
5.2	Process Related Development Needs	57
5.3	Communication Related Development Needs	59
5.4	Requirements Related Development Needs	60
5.5	Summary of Inputs for the Initial Proposal	60
5.6	Initial Proposal for Improvements to the Development process of a Standardized P2P system	62
6	Validation of the Proposal	67
6.1	Overview of the Validation Stage	67
6.2	Feedback to the Initial Proposal for Improvements to the Development Process of a Standardized P2P System	67
6.2.1	Feedback and Developments to Temporary Feature Teams and Cross-functional Feature Teams	69
6.2.2	Feedback and Developments to Keeping the Sprints Time-boxed and Adjusting Sprint Timing	69
6.2.3	Feedback and Developments to Sprint Reviews	70
6.2.4	Feedback and Developments to Sprint Planning	70
6.2.5	Feedback and Developments to Daily Scrum in All Teams	70
6.2.6	Feedback and Developments to Allocate Time for Code Review and Dedicated code reviewer(s)	71
6.2.7	Feedback and Developments to Product Owner into Retrospectives	71
6.2.8	Feedback and Developments to Criterion for getting into Sprint backlog and Test cases as requirements	71
6.2.9	Feedback to Provide Training of Requirements	71
6.3	Final Proposal for Improvements to the Development process of a Standardized P2P system	72

6.4	Actions Approved Immediately for Implementation Based on the Final Proposal	73
7	Conclusion	75
7.1	Executive Summary	75
7.2	Next Steps and Recommendations toward Implementation	77
7.3	Thesis Evaluation	77
7.4	Closing Words	79
	References	1

Appendices

Appendix 1. Initial interview questions to direct the interviews

Appendix 2. Summary of main content points from Interview 1 (example)

Appendix 3. Summary of main content points from Interview 3 (example)

Appendix 4. Jamboard page 2/5 from interview 3

List of Figures

- Figure 1. The research design of this thesis.
- Figure 2. The Product Development operation's teams and their responsibility areas.
- Figure 3. An overview of the current Product Development process.
- Figure 4. Sprint relations in and between teams.
- Figure 5. Sprint lifecycle management process.
- Figure 6. Ticket lifecycle.
- Figure 7. An overview of the Product Development process after corporate changes.
- Figure 8. A sprint management process.
- Figure 9. A sprint management process - two week sprint.
- Figure 10. Sprint relations after corporate changes.
- Figure 11. Relations between Roadmap, Feature, TFS item and Tasks.
- Figure 12. Agile Principles (Shore and Warden 2021: The Agile Manifesto).
- Figure 13. Scrum framework (Rubin 2012: Scrum activities and artifacts).
- Figure 14. Two products and multiple component teams (Rubin 2012: Feature teams versus component teams).
- Figure 15. Combined feature team and component teams (Rubin 2012: Feature teams versus component teams).
- Figure 16. Adjusting Sprint timing.

List of Tables

- Table 1. Details of Data collections 1-3 used in this study.
- Table 2. Development needs that were fixed by the corporate changes or otherwise ruled out of scope of this study (based on the current state analysis).
- Table 3. Development needs considered as focus areas of this study (based on the current state analysis).
- Table 4. Conceptual framework of selected best practice elements to guide the Agile processes improvements in this thesis.
- Table 5. Key stakeholder suggestions (findings of Data 2) for Proposal building in relation to findings from the CSA (Data 1) and the conceptual framework.
- Table 6. Initial Proposal of this thesis.
- Table 7. Evaluation of the Initial proposal.
- Table 8. Final Proposal of this thesis.

1 Introduction

The case company of this thesis provides customizable and standardized Procure-to-Pay (P2P) system solutions. The product development operations, this thesis deals with, comprise them both. However, this thesis explores the development possibilities of the process that covers the product development of a *standardized P2P system* from the initial change requests to the version release.

This thesis sets out to find the ways how to improve this process in terms of time, reduction of overlapping tasks, and in relation to other functions within the Product development operations. As known from practice, the operations will not perform as well as they could, unless their individual processes are well designed (Slack and Brandon-Jones 2019: 177).

1.1 Case Company and Business Context

The case company of this thesis is a provider of financial process automation solutions, both customized and standardized. The case company employs approximately 80 employees and its global organization has headquarters in Ireland and offices also in UK, USA and the Nordics.

The product, Procure-to-Pay system, automates the finance process from procurement to invoice automation and payment posting via an integration to the customer's financial or ERP system and finally stored in the Enterprise Content Management system. The P2P system is offered as a single-platform solution which contains all the mentioned functionalities in one interface. There are two ways to deliver the product to customers; a standardized system as it is, and a customized system with unique queries, dashboards, notifications and integrations. The offering also includes add-in solutions, such as gathering invoices from email to the P2P system, scanning invoices to the P2P system, and more.

The Product development operations for the Procure-to-Pay system cover all the offered solutions and all aspects of those solutions, such as back-end, front-end, integrations, configuration, automated tests and manual testing. There are three main solutions; standardized P2P system, customized P2P system, and the ECM. The Development

processes of these main solutions are highly linked to each other and are addressed in the current state analysis of this study.

1.2 Business Challenge, Objective and Outcome

The Product development operations are vital for the case company's business to address the customers' needs and, most importantly, to support the sales to have continually working system with updated information security measures and with cutting edge technology and features to boost the sales.

The standardized P2P system is a Procure-to-Pay system. The standardized P2P system includes initiating a purchasing request to approval, placing a purchase order, delivery and receipting of the goods, receiving and approving the invoice, integrations to ERP for payment details and archiving the document. The standardized P2P system is more effective for new customers to take into use since it has all the configurable settings and follows the general use cases. The standardized and customized systems share the same code, for the back-end and the front-end. The difference is in the configurable settings, which are missing from the customized system when a new customer starts an implementation project with the case company's consultants. In these projects, the consultants configure the setting for the specific customer use, and they rely on the documentation of the configurable settings that are created during *the Development process of the standardized system*.

The business challenge of this study addresses to the Development process of the Standardized P2P system. The business challenge was identified by the case company's R&D Director as:

The Development process of a standardized P2P system has some bottlenecks and hindrances. They need to be identified, and the process improved. (R&D Director)

These bottlenecks and hindrances prolong the Development process of the standardized system, although they stay unknown. The goal for the thesis was to identify them and proposed improvements to tackle them. A new version of the system, which gathers all the changes done to the code, to the ECM's code, and to the configuration settings, is released monthly. The correlation of the standardized and customized systems and the ECM, their linked Development processes and shared resources, e.g. documentation

and testing personnel, are affected negatively, when the Development process of the standardized system is lagging behind.

Accordingly, the objective of this thesis is *to propose improvements to the Development process of a Standardized P2P system.*

The outcome of this thesis is *the proposal for improvements to the Development process of a Standardized P2P system.*

1.3 Thesis Outline

The scope of this thesis is the Development process of the standardized system, from the initial change request to the version release. The Product development operations in total are covered in the current state analysis in order to illustrate the understanding of the relations between different processes. The research is conducted as a qualitative action research by using methods, such as internal documentation, interviews and workshops, and observations in internal training.

The research is organized as follows. Firstly, the current state of the development process of the standardized system and the product development operations are analyzed by conducting interviews with stakeholders, who have different responsibility areas. Additionally, observations are gathered in continuous meetings, such as daily scrum meetings. Thus, the development needs of the development process are identified. Secondly, the existing knowledge and best practices are studied and a conceptual framework for process improvement is generated. Thirdly, the proposal of the needed improvements is built. Finally, the proposal is finalized after validation.

2 Method and Material

This section introduces the research approach, research design and research quality criteria of this thesis and describes the data collection and analysis methods.

2.1 Research Approach

According to Baxter, Hughes and Tight (2014:59), *methodology* refers to the approach or paradigm that underpins the research, while *method* relates to the tools of data collection or analysis. Baxter et al. (2014:64) present the methodology to be defined by research family and research approach.

Research family can be either quantitative or qualitative and either deskwork or fieldwork. There are four research approaches especially popular in business and management studies: action research, case study, experiment and survey. Research methods or techniques are the following: documents, interviews, observations and questionnaires. Baxter et al. (2014: 67) also underlines that this classification is neither definitive nor exclusive, and individual projects may involve more than one of these approaches. Neither the linking between families, approaches and techniques is written in the stone, but there are some indications. For example, experiments tend to be quantitative in nature and action research tends to imply fieldwork. It is up to the researcher, their preferences, available resources, the constraints and the particular issues which will be researched, whether different research families, approaches and techniques will be used in combination (Baxter et al. 2014: 67-68).

According to Coughlan & Coughlan (2002: 222-223), Action research focuses on research *in* action by using a scientific approach to study the resolution of important organizational issues together with those who experience these issues directly. Action research also involves the members of the system being studied in order to make the action studied more effective while building up scientific knowledge. Finally, action research – being an iterative cycle of data gathering, analysing, action planning, taking action and evaluating – is an approach to problem solving, an application of the scientific method of fact finding and experimentation to practical problems (Coughlan & Coughlan 2002: 222-223).

Thus, the research family of this thesis is qualitative fieldwork with action research as the selected research approach. Since the purpose of this thesis is to propose changes to a process by researching, understanding and analysing, the research techniques used in this thesis include a defined set of interviews, workshops and documentation. Due to the data collected comprehending the participants' personal views and thoughts rather than explicit statistics, the type of data is qualitative. The interviews are semi-structured leaving room for open discussion.

2.2 Research Design

The research design of this thesis is split into several stages and describes the data collection and outcomes as can be seen in Figure 1.

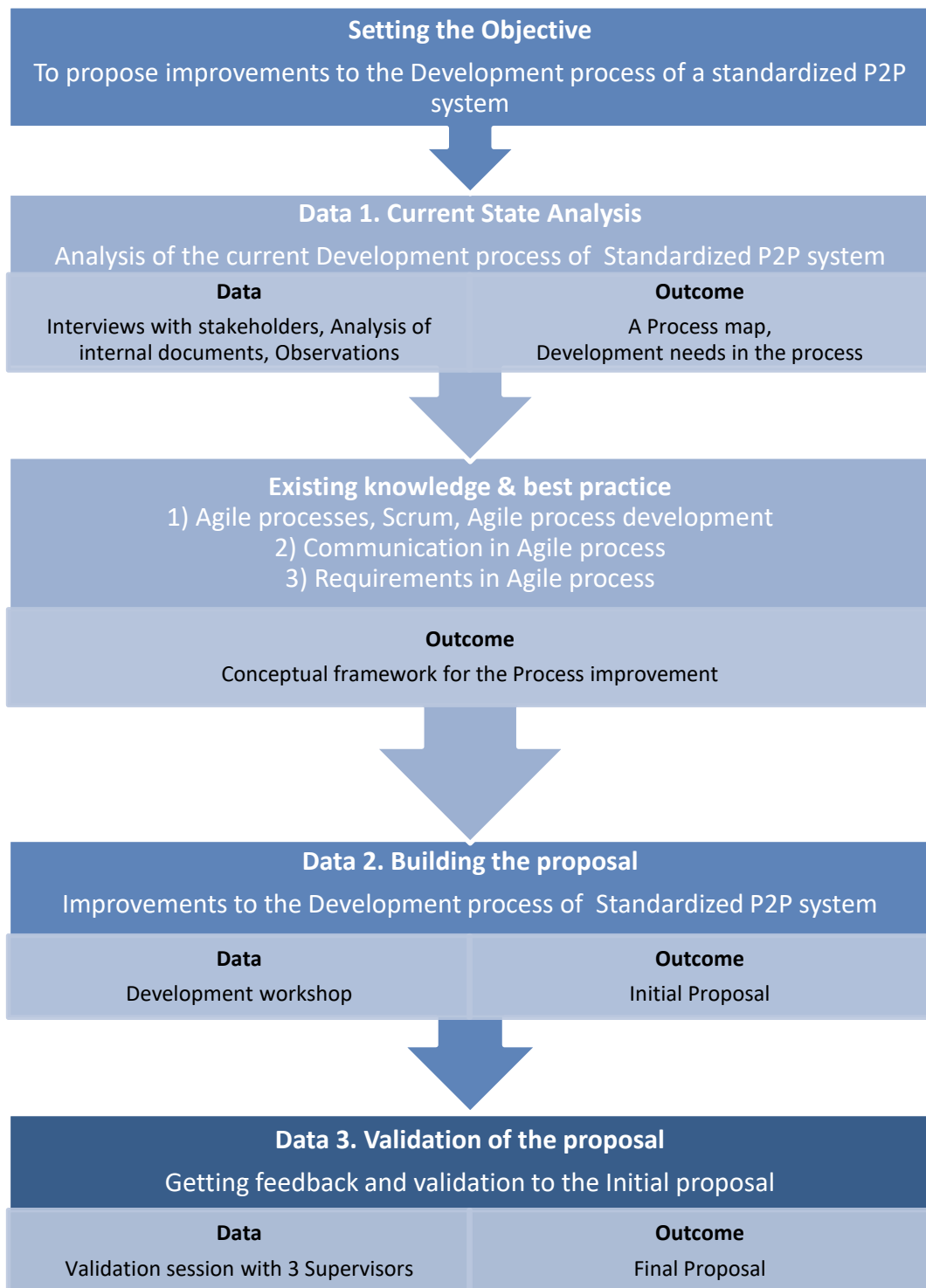


Figure 1. The research design of this thesis.

As Figure 1 shows, the study starts by setting the objective. The next step is the current state analysis where the current practices are analyzed in the Product development operations and processes. In addition, the relations between different processes are described in order to understand the context of the Development process of the

standardized P2P system. Finally, the scrutinized process is mapped and the development needs are identified. Data is gathered by conducting interviews, by making observations in continuous meetings and by analyzing the internal documentation. The current state analysis findings inform the scope and focus for the next part of the study.

The next part of the study focuses on reviewing the existing knowledge and best practices in process development, with special attention to agile environment. The goal is to learn about best practice in process improvement specific for the findings of the current state analysis, and to develop a conceptual framework to guide the following parts of the study.

Based on the results of the current state analysis and suggestions selected from literature that are merged into the conceptual framework, the initial proposal is built to suggest improvements to the Development process of the standardized P2P system. The proposal is built by a project team from the case company involved in the creation process in a form of a workshop. An initial proposal is drafted based on the conceptual framework and the input from the workshop.

Finally, the proposal is validated in a specific session by three supervisors; R&D Director, Head of Solutions team and QA Manager. Based on their feedback, the proposal for improvements to the Development process of the standardized P2P system is revised and finalized for practical implementation.

2.3 Data Collection and Analysis

In this study, data is collected from a variety of data sources in order to get evidence for the current state analysis, the proposal building, and the validation. The data collection details are presented in Table 1.

Table 1. Details of Data collections 1-3 used in this study.

	Participants/ role	Data type	Focus	Date, length	Documente d as
<i>Data 1, for the Current State Analysis (Gate 3)</i>					
1	Respondent 1: R&D Director	Skype meeting	Focus on overall picture of Development Operations	April 7, 2020 1,5 hours	Field notes, Jamboard and recording
2	Respondent 2: Head of Solutions team	Skype meeting	Focus on the development process of the standardized P2P system	April 15, 2020 1,5 hours	Field notes, Jamboard and recording
3	Respondent 3: QA Engineer	Skype meeting	Focus on quality assurance processes	April 16, 2020 2 hours	Field notes, Jamboard and recording
4	Respondent 4: Senior Software Engineer	Skype meeting	Focus on development process of P2P system (Java)	April 17, 2020 2 hours	Field notes, Jamboard and recording
5	Respondent 5: Software Architect (ECM)	Skype meeting	Focus on development process of ECM system	April 17, 2020 2 hours	Field notes, Jamboard and recording
6	Respondent 6: Quality Manager	Skype meeting	Focus on overall picture of Development Operations and on quality assurance processes	April 20, 2020 2 hours	Field notes, Jamboard and recording
7	Respondents 1 & 2	Face-to-face meeting	Focus on overall picture	August 13, 2020 1,5 hours	Field notes
8	Java, Solutions and QA teams	Scrum meetings	Work done and work to-be-done	Every day 0,5h	N/A
9	Java, Solutions and QA teams	Retrospective meetings	Evaluation of the latest sprint	Once in two weeks, 1-1,5 hours	Retrospective documentation
<i>Data 2, for Proposal Building (Gate 5)</i>					
10	Participants: Head of Solutions team, Senior Software Engineer, QA Engineer	Workshop via Teams	Building the initial proposal	October 27, 2020 2 hours	Field notes and recording
<i>Data 3, for Validation (Gate 6)</i>					
11	Participants: R&D Director, Quality Manager, Head of Solutions team	Validation session face- to-face & via Teams	Validating the initial proposal	November 10, 2020 1,5 hours	Field notes and recording

As seen from Table 1, data for this thesis was collected in three rounds. The first round, collecting Data 1, was conducted for the current state analysis and six respondents were selected for interviews according to their position and expertise. Additionally, observations were gathered during daily scrum and retrospective meetings. The goal was to describe and map the Development process of the standardized P2P system and the context it is in, meaning the Product development operations and all the processes within and their relations to each other. When the Development operations and especially the scrutinized process is mapped, the development needs are pinpointed.

In the next round, Data 2 was collected in order to formulate the initial proposal of improvements to the Development process of standardized P2P system. A workshop was facilitated for the purposes of gathering suggestions and viewpoints for the process improvement. Participants for this workshop were selected according their expertise and relation to the scrutinized process.

In the third round, Data 3 was collected in an organized validation session. The validators are from managerial level and they all have a role in the scrutinized process. Data 3 includes validators' feedback to the initial proposal.

The primary method of data collection in this study was interviews, which were conducted in a semi-structured way by leaving room for open discussion. The questions were created and shared with the interviewees in advance. The interviews were held via Skype and recorded for the purposes of revisiting them during this research. In addition, field notes were taken and a post-it process map was drafted with Jamboard during the interview. The questions for current state analysis interviews can be found in Appendix 1. The summaries of field notes from interviews 1 and 3 can be found in Appendices 2 and 3. Jamboard page 2/5 from interview 3 can be found in Appendix 4.

Continuous meetings, such as daily scrum and retrospective meetings, provided critical and profound information of the process and its development needs for the current state analysis. The observed information was captured as field notes. Also, internal documentations was analyzed for the process mapping and development needs analysis.

The workshop for building the initial proposal was conducted via Teams. There was room for all participants to describe how the pinpointed development needs could be improved and to give feedback of the conceptual frameworks improvement suggestions.

The validation session was held via Teams and participants included the R&D Director, the Head of Solutions team and the QA Manager. The findings of the current state analysis and the initial proposal for improvements were presented for validating in terms of practical implementation.

Thus, the biggest part of the data was gathered and analyzed for the current state analysis, to establish the understanding of the Development operations, the scrutinized process and its development needs. The findings from the current state analysis are elaborated in Section 3 below.

2.4 Research Quality Criteria for This Thesis

As this thesis uses action research and qualitative research methods, according to Saunders, Lewis and Thornhill (2016: 206-207), the research quality criteria for an action research include dependability, credibility and transferability. Dependability means producing dependable account of the research focus, which may be modified as the research progresses, in order to be understood and evaluated by others. Credibility puts emphasis on ensuring that the research participants' socially constructed realities corresponds in detail to what the participants intended. Transferability provides the reader with an opportunity to judge whether the study can be transferred to another setting in which the reader is interested by providing a full description of the research questions, design, context, findings and interpretations (Saunders et al. 2016: 206-207).

In addition, a rigorous action research has a relevant and clear research question, a well-structured and documented data collection process to an adequate and rich description and a logical analysis (Näslund, Kale, Paulraj 2010: 339). This means that methods and procedures are explicitly described with significant levels of detail. The methods and procedures should show the "audit trail" of how data were collected, processed, condensed or transformed, and displayed and the conclusions are clearly linked with exhibits of displayed data. According to Näslund et al. (2010: 332), Action research can provide significant research contributions for both practice and academia, when a rigorous methodology is adopted.

This thesis aims at strengthening dependability by, first, conducting a focused current state analysis that constantly checks the relation to the business problem of the case company, and then recording all the changes and steering a focused study of existing knowledge guided by the current state analysis findings to build the conceptual framework that informs further development steps. By following this approach, the entire study – from setting the objective to the final proposal – is planned to be well documented, grounded in empirical evidence, and keeps high relevance to the practical challenges of the case company.

In this thesis, credibility is planned to be ensured by relying on extensive documentation analysis, conducting participant observations, and involving the most relevant and knowledgeable key stakeholders in development actions of this study. In addition, the research design and data collection plan are planned ahead and in detail, while should be revised and corrected according to the intermediate findings of the study, if needed. Data collection is planned to ensure triangulation via the analysis of the case company's internal documents, interviews and observations. Data is captured in multiple forms; recordings, Jamboard, files, and field notes.

Finally, regarding transferability, this thesis is planned to be conducted and presented in a transparent way in order to make the research process open to other teams, and thus help the results to be used also in other processes of the case company. The research design and objectives, explicit reports on the findings and interpretations, openly available existing knowledge and the conceptual framework should all help to make the results applicable for any process improvement in research settings.

3 Current State Analysis of the Product Development Operations and the Development Process of the Standardized P2P System

This section discusses the current state of the Development Process of the Standardized P2P System analysis and the recent changes initiated by the case company. While the study focuses on improving a single Development process, the analysis also elaborates the relations and dependencies to the other concurrent processes.

3.1 Overview of the Current State Analysis

The goal of the current state analysis is to document and analyze the Product Development operations, and more specifically focus on the Development Process of Standardized P2P System, pointing out the development needs of this process. Importantly, during the course of this research, the case company presented changes to the Product Development operations, including the Development Process of the Standardized P2P System. The current state analysis reports both situations, before and after the changes to the scrutinized process, and analyzes them both.

Firstly, the current state starts with the relations between different teams in the Product Development operations *before* the corporate driven changes. Secondly, the overview of the Product Development operations is presented. Thirdly, the overview is broken down into more detail in terms of sprint lifecycle, ticket lifecycle and concurrent sprints. Fourthly, the current state of the Product Development operations *after* the corporate driven changes is given. Lastly, the development needs of *the Development Process of Standardized P2P System*, with some attention to the context (i.e. the wider Product Development operations), are pointed out *after* the changes.

The corporate driven changes were initiated after the start of this study. The implementation of the changes and their impacts were closely observed and documented, and incorporated into the current state analysis.

3.2 Description and Analysis of the Current Product Development Operations

Initially, the ECM was the only product that the case company offered, but the Development of the P2P system has started in 2014. Once the P2P system was up and running, the product development had been mostly driven by the requests of one enormous project. During the last year 2019, the Development operations have evolved

to its current form of functioning. Once the enormous project was ready, the balance has shifted more to overall development and since then the R&D Director's aim has been to get all personnel of Product Development Operations to work in the same country and preferably under the same roof. The Product Development Operations involve couple of dozen employees and they used to locate in Ireland, Spain, Belarus and Finland.

3.2.1 The Current Product Development Process and Teams

In the case company, the agile Product Development Operations consist of four teams: ECM, Java, Solutions and Quality Assurance. The first three teams conduct different implementations in order to develop the P2P system, and the fourth team conduct the quality assurance.

The first three teams and their responsibility areas combined produce the full P2P system. While these teams develop the system, they are also responsible of the testing of the developed functionalities. Figure 2 illustrates the relations of these teams and their responsibility areas.

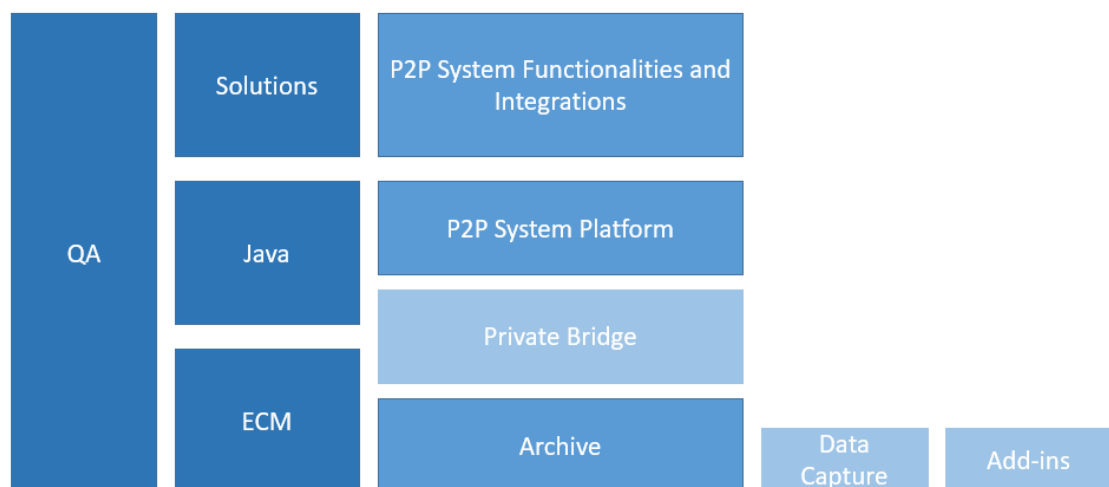


Figure 2. The Product Development operation's teams and their responsibility areas.

The first team, ECM (Enterprise Content Management) is responsible of the ECM system solution. ECM is a stand-alone system which can be sold as a single product for the customers. However, the ECM is also a part of the P2P system and is connected via Private Bridge integration to the platform of the P2P system. The ECM team is also

responsible of other Add-in solutions the case company provides, for example, Data Capture of invoices from an email.

The second team, Java, develops the platform of the P2P system. The platform itself does not provide any functionality for the customers, but it provides the ability to have any functionality. The platform provides the basis for the standardized and customized systems.

The third team, Solutions, is responsible for applying configurations to the platform. The configurations set the platform provided functionalities in use. The Solutions team apply the configurations for the standardized system. For the customized systems, the project team's consultants apply the customer specific configurations and they typically use the latest version of the standardized system as the basis. The Solutions team's documentation is used by the consultants when a customer is upgrading their P2P system from an older version to the newest or some new functionality is added.

The fourth team, Quality Assurance, is responsible for testing of all newly developed functionalities from all the three teams before anything will be released, but they conduct the testing from the business' and customer's point of view. The QA team is also responsible of regression tests; assuring any old functionalities will not be broken due to the newly developed ones.

The content of what functionalities these teams are working on is provided from different sources and is planned on different levels. The overview of this process is illustrated in Figure 3, with different colors indicating responsibilities.

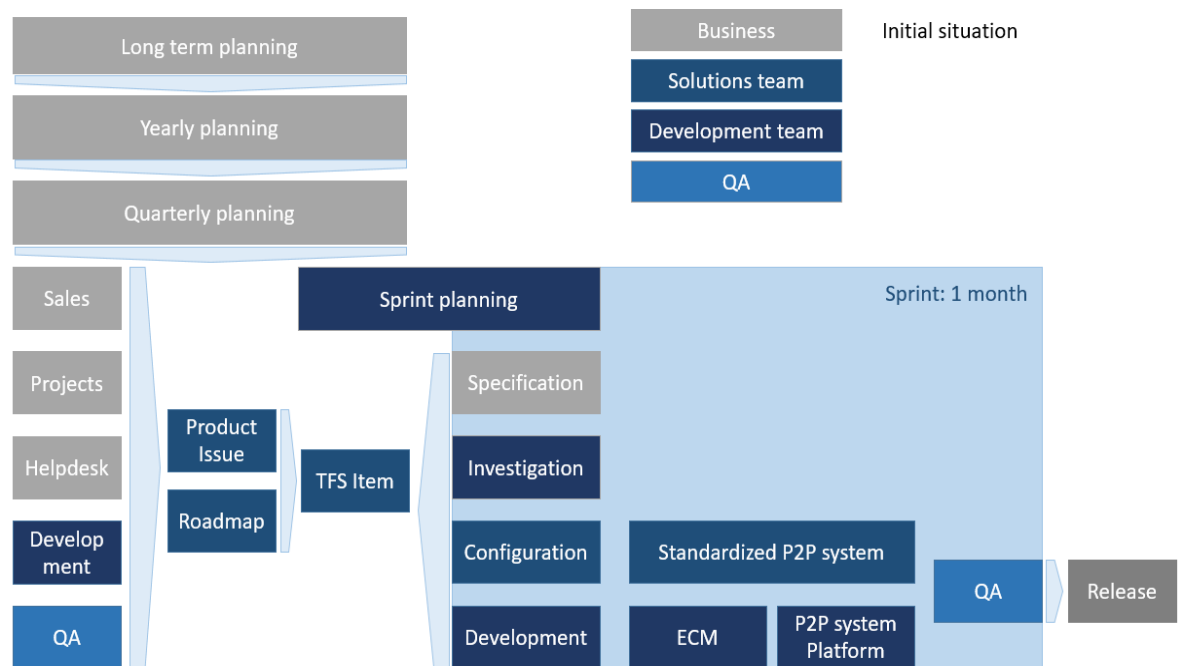


Figure 3. An overview of the current Product Development process.

As mentioned earlier, presently the Development of P2P system is driven by requests. As seen from Figure 3, requests for changes may come from five sources: Sales, Ongoing projects, Helpdesk, Product development, or the QA team. If the people working in these areas come across an issue or a new desirable feature, they send a detailed email to Product Issue or to Roadmap mailbox to be evaluated. These requests will be logged as work items, tickets, to the Microsoft Team Foundation Server, where they will be evaluated, allocated, planned, assigned and reported ([Sender] 2020: Received email in Roadmap mailbox, [Sender] 2020: Received email in Product Issue mailbox).

The long-term planning process was described as follows:

The long-term planning, yearly and quarterly planning is conducted by the sales and business management. This planning is based on market, competitor and customer research. (R&D Director)

The planning process produces new emails to roadmap mailbox and thus new TFS items, tickets.

The tickets are categorized into four kinds of tickets. *Specification tickets* are for requests needing detailed planning from the business point of view. This is usually conducted by

the Chief Technology Officer. *Investigation tickets* are for requests which need to be planned from the technical point of view and these are most of the times investigated by the R&D Director. *Configuration tickets* are the ones the Solutions team is working on. *Development tickets* are for both ECM and Java teams and are broken down in multiple child tickets more in detail based on the technical requirements ([Case company] 2020: Data queries).

Moreover, the TFS items combined is a ticket backlog and the planning of which of the items end up in the final product and when, is conducted in *the Sprint planning*. A sprint is a one month long period, but the sprint planning process is conducted in weekly meetings. Where, according to the R&D Director, all new TFS items are validated and allocated either to the current sprint or to the future sprints by the CTO, R&D Director and the Head of Solutions team.

Since all the different sections of the P2P system are interlinked, some of the bigger items may need, first, implementation to *the ECM* by the ECM team, then changes to *the P2P platform* by the Java team, and lastly configuration to *the Standardized P2P system* by the Solutions team. This does not apply to all work items, but it has an impact on the sprints under development in each team ([Case company] 2020: Data queries).

After ECM, Java and Solutions teams have exported their versions for release, the QA team tests and validates the features and runs the regression tests. Only after QA's validation the final releasable version is shared in-house, sales and project personnel may use it.

3.2.2 Description of the Current Sprints and Ticket Lifecycles

The logic and relations of the sprints in and between teams are shown in Figure 4. The relations are especially important for implementing big TFS items, that result in changes to the ECM, P2P Platform and finally to the Standardized P2P System.

	Sprint 1	Sprint 2	Sprint 3
Release	N	N+1	N+2
QA	N	N+1	N+2
Solutions	N	N+1	N+2
Java	N+1	N+2	N+3
ECM	N+2	N+3	N+4

Figure 4. Sprint relations in and between teams.

Figure 4 shows that release N+2 will be finished at the end of Sprint 3 and the configuration by the Solutions team and testing by the QA team is conducted during that same sprint, in the same month. However, the Java team has done their release N+2 in the previous month, so the platform changes needed for some configuration items are already in place and ready for the handover. In addition, the ECM team has completed their release N+2 already a month before that to have the needed ECM changes available for the Java team ([Case company] 2020: Internal documentation of working items included in the release).

The sprint content is bound to time. This way the amount of worked TFS items may increase during the sprint and decrease at the end of the sprint ([Case company] 2020: Data queries). The sprint lifecycle management process is presented in the following Figure 5.

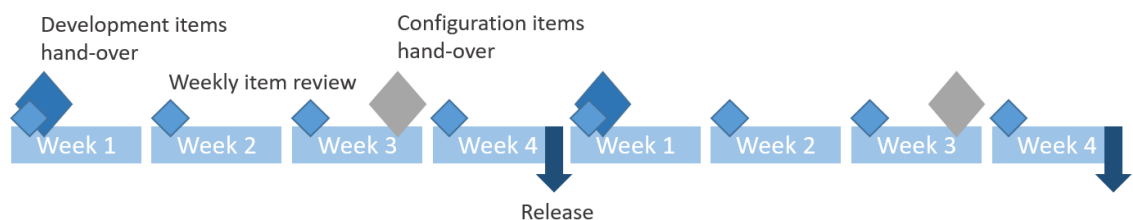


Figure 5. Sprint lifecycle management process.

As seen in Figure 5, in the beginning of every sprint, a knowledge transfer meeting is held where the development items from the last sprint implemented by ECM and Java are presented to the Solutions team. The meeting is for getting to know ahead what items have been implemented and what kind of configuration do those require. When the Solutions team is close to delivering the newest version for QA team for testing, another knowledge transfer meeting is held to provide the information of the sprint content to QA team.

When the sprint content is planned, allocated and assigned, TFS items share the needed information. All TFS items have a parental item, a Feature, which divides the items into bigger sections, such as ECM, P2P, Portal, etc. The TFS items provide the information of the request source, requirement, implementation and success criteria, but also the estimations and actual working hours of implementation and testing ([Case company] 2020: Data queries).

When the TFS items are ready for implementation, the status changes are recorded. The ticket lifecycle is illustrated in the following Figure 6 for the configuration and development items.

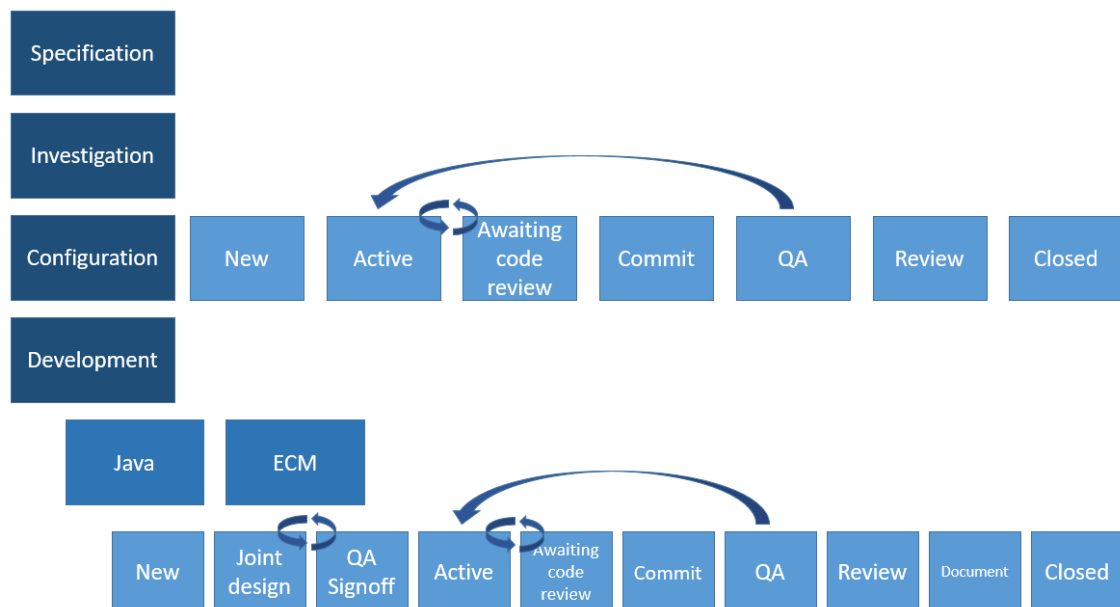


Figure 6. Ticket lifecycle.

As Figure 6 shows, initially when a new TFS item is created, it gets the status *New*. The full path of the tickets is visible in *Development* TFS items which are used in Java and

ECM teams. Once the ticket is created, it gets a status *Joint design*, which is a step for more elaborated planning for the engineer and the system architect. Then, the ticket moves to *QA Signoff* where a dedicated QA team member evaluates the ticket content if it is understandable and all aspects have been considered. If the ticket needs some additional information, it is returned back to the status *Joint design*. When the ticket is in the implementation, it gets a status *Active*. Once the ticket content has been implemented in the Development environment, it gets sent to the status *Awaiting code review*. All tickets are peer reviewed for validation of the implementation quality and extent. If something is missing or could be implemented in a more efficient way, the ticket is returned back to the previous person with comments and status *Active*.

However, with the *Configuration* tickets, the two statuses, Joint design and QA Signoff, are not currently used. For configuration tickets, the Active status includes also producing the documentation of what has been configured. *Commit* status for *Configuration* tickets means implementing the documented and reviewed configurations to an isolated environment with no data at all, only the settings.

For Development tickets, the *Commit* status means bounding the code to a branch, which can be later tied to a version release. After committing the implementations, they are shared for QA team to test from a business and end-user point of view. The tickets are handled by QA team as follows:

Ticket might be returned back to the implementer if it does not meet the requirements and the priority is so high it will affect to the whole release. If the found issue is not urgent, the tester will send the description of it as an email to Product Issue and a new ticket will be created for a future sprint. (QA Engineer)

The manager of the QA team will then *Review* all documented tests, if the extent was appropriate. A dedicated member of the QA team is also responsible for updating the internal manual, which is used by all teams in the Product Development operations and the project consultants. This is why the development item may have a status of *Document* before it is *Closed* if the ticket is adding the functionality abilities to the platform.

3.3 Development Needs of the Current Product Development Process

Based on the interviews, observations and document analysis, the following development needs were identified in the current Product development process. Importantly, these results were obtained *before* the corporate changes were initiated in spring 2020. The results after the corporate changes are discussed in the next section.

Biggest issue from the business perspective is the length from a requisition of a change or a fix to the actual delivery to customer's environment. The process takes several months:

Firstly when the initial request from the customer comes it may take a month before the work will be started. If the item requires both development and configuration the working time takes two months. After possibly three months the item can be implemented to customer's testing environment for acceptance testing. This is way too long cycle to get the items to customer environments and it forces us to make fixes to old released versions which has risks in terms of quality. (R&D Director)

Another big problem for business is updating the environment configuration-wise to the newer version. Upgrading the ECM and P2P platform have been implemented as semi-automatic system updates, but the upgrade of the configurations is conducted manually based on the Solutions team's documentation. According to the QA Manager, this is a big reason for customer environments having a year or two old versions, since the amount of the configurations applied is massive and it would require several unbillable working days for consultants. However, this has been recognized and the planning and implementation of the solution has been started.

The researcher observed from the interviews that no other person than R&D Director was capable of answering or knew where and how the TFS items come from. Employees have no visibility to the planning process, only to one's own work.

The R&D Director and the Head of Solutions team agreed that, firstly, the sprint planning should cover larger entities and proactive development rather than going through individual defects and fixes. Secondly, the sprint focus is not clear and it should be mainly influenced by the long-term planning process.

The R&D Director states that there has been issues with the quality. Multiple participants described the issue to be too vague requirements and specification of the expected outcome. This leads to solutions which do not solve the initial problem or may create some new ones.

Multiple issues related to ticket management were raised by QA Engineer, Senior Software Engineer and Quality Manager.

Sometimes we don't have the technical TFS item and we are coding based on the business ticket. This means that intended planning and validation from the technical perspective is lacking. Other times we might have the technical ticket, but the business ticket is missing. Also a step between QA Signoff and Active would be needed to indicate the item is waiting to be worked on. Additionally, sometimes the needed development in the ECM side is in planning phase when we would already need it. (Senior Software Engineer)

Some items are tested twice in QA, because first we test the development item from Java team and after Solutions team has configured on top of that development, we test it again. (QA Engineer)

When the sprint is closing, some TFS items might be still in process even though those should be in the status QA in order the whole version could be tested by QA team. (QA Manager)

It was also mentioned by the R&D Director and the QA manager that as QA team needs to update the test environment to the latest version, all the pre-existing data disappears. It costs time and money to build the environment in each sprint and in addition, the QA team doesn't have a large environment with loads of data and users, which would present the customer's environments.

Lastly, the lack of communication within teams, departments and the whole organization was raised by the Head of Solutions team and other interviewees.

However, these results were obtained *before* the corporate changes were initiated. The company, being agile and seeing the need for improvements, especially in terms of shortening the sprint time, has initiated changes to the Product development process that are reported and analyzed below.

3.4 Description and Analysis of the Development Operations after Corporate Changes

In spring 2020, the business management of the case company presented changes to the Product Development Operations. The changes were based on business' lack of visibility to Product Development Operations and development items and the business management wanted to shift the sprint content planning to be driven by sales and business management. In addition, new development items wanted to be released faster.

3.4.1 Descriptions of the Development Process Operations and Teams after Corporate Changes

Figure 7 presents the Product development process from planning to release after the corporate driven changes.

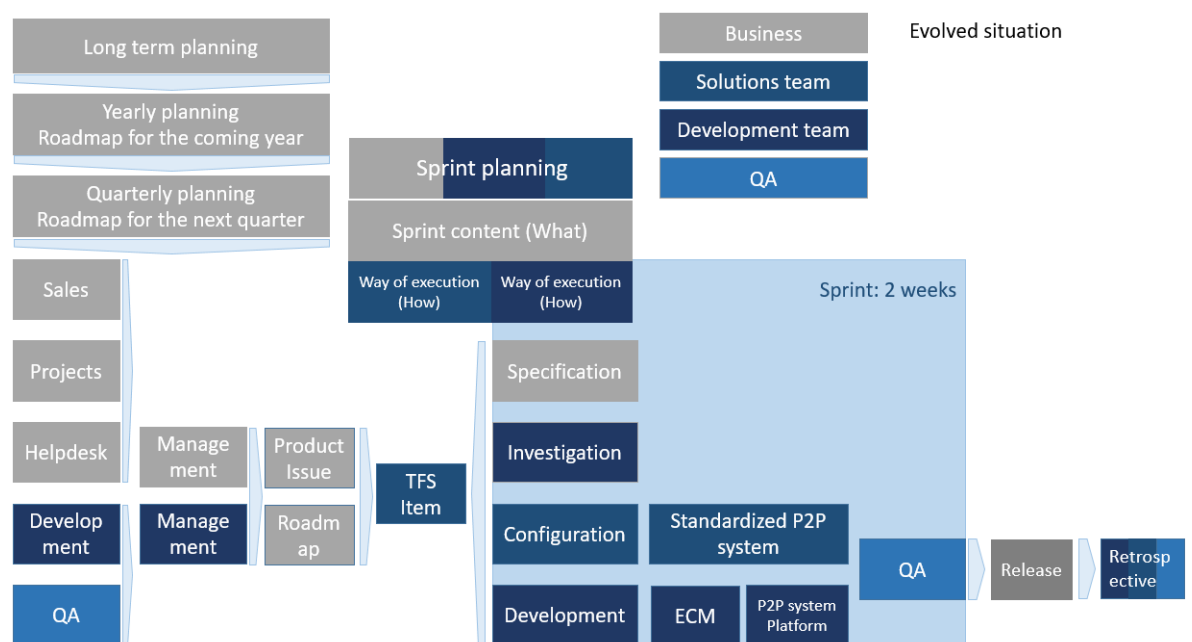


Figure 7. An overview of the Product Development process after corporate changes.

As seen in Figure 7, the biggest impact of the changes were requests handling, sprint planning and sprint length. Firstly, change requests and product issues wanted to be filtered by the management before the TFS items were created. Emails are sent to dedicated managers instead of directly to the two mailboxes. Manager's then review the requests and pass those forward to the Product Issue and Roadmap mailboxes if the

requests are relevant ([Sender] 2020: Received email in Roadmap mailbox, [Sender] 2020: Received email in Product Issue mailbox).

Secondly, sprint planning is conducted together with business, Head of Solutions team and R&D Director. Content wise, sprint planning is conducted by the business management and especially sales. The content planning is based on customer reviews and market and competitor research. After the business has established the scarce plans of what should be developed and when, the Head of Solutions team and R&D Director review the content from a technical and resources point of view. Their input determines how and when the content will be delivered with the current resources.

Thirdly, for continuous improvement, learning and knowledge sharing after the process has changed in a large scale, Retrospectives were taken into use. After each sprint the Product development operations recall what was done well and what can be improved in that sprint by the lead of a dedicated QA team member who documents the discussion ([Case company] 2020: Memos).

Lastly, due to the faster deliveries of the new development items, the sprint length was reduced from a month to two weeks. This introduced multiple practical changes that are elaborated in the next chapter.

3.4.2 Descriptions of the Sprints and Ticket Lifecycles after Corporate Changes

Corporate changes were also introduced to the sprints and ticket lifecycles. A month long sprint and the recurrent meetings from the Solutions team's point of view is illustrated in Figure 8 for comparison.

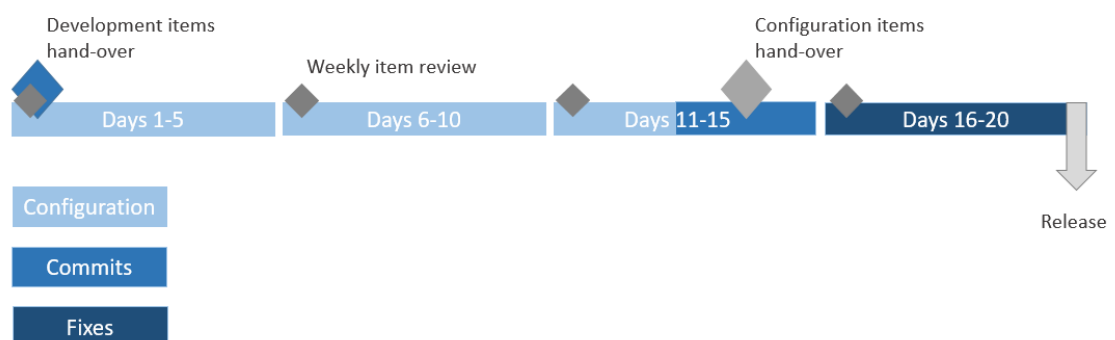


Figure 8. A sprint management process.

As Figure 8 shows, with a month long sprint there were two and a half weeks for configuration, couple of days for committing the new items to the isolated environment and one week to implement fixes based on QA team's tests and comments. The following Figure 9 illustrates a two week long sprint and the recurrent meetings from the Solutions team's point of view.

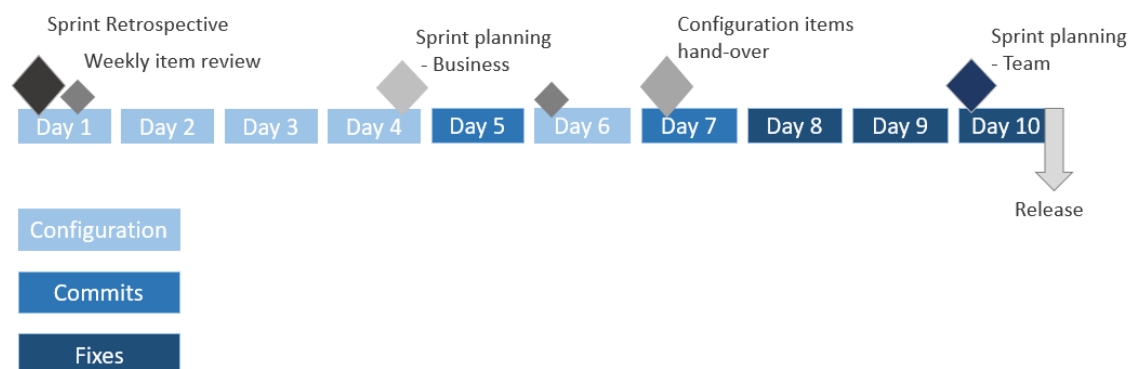


Figure 9. A sprint management process - two week sprint.

As seen in Figure 9, the two weeks are practically ten working days. As the R&D Director described the new process; There are five and a half days for configuring the items, a half day of committing the fully ready items at the end of first week and a full day of committing all completed items on the seventh day. A release is delivered to the QA team for testing in the end of the seventh day. Three last days are for fixes based on QA tests and comments, and for planning the next sprint.

TFS item reviews are conducted on a weekly basis as earlier, but a sprint planning meeting is held only once in two weeks by business, Head of Solutions team and R&D Director and separately within Solutions team. Team's own sprint planning meeting is a new practice and helps the team members to fully understand the items and prepare questions for the business if needed. Also the items are allocated for specific team members during the planning meeting.

Sprint retrospective is held after the QA team has validated the release and it has been handed over the internal use. From Solutions team's point of view this happens in the beginning of each sprint ([Case company] 2020: Memos).

Configuration items are handed over to the QA team right before the version is released for testing, but there is no handover of Development items anymore. This is due to the shortened time, the development items which require configuration are tested by the QA only after the configurations are in place as well. Sprint relations were shifted in a way that all teams are working on the same version during the same sprint as the following Figure 10 illustrates ([Case company] 2020: Internal documentation of working items included in the release).

	Sprint 1	Sprint 2	Sprint 3
Release	N	N+1	N+2
QA	N	N+1	N+2
Solutions	N	N+1	N+2
Java	N / long term	N+1 / long term	N+2 / long term
ECM	N / long term	N+1 / long term	N+2 / long term

Figure 10. Sprint relations after corporate changes.

As Figure 10 shows, all the teams are working on the same version during the same release.

In addition, there are bigger functionalities that can't be completed during the two weeks sprint and those are called long term development. Java and ECM teams might be developing a functionality during multiple sprints but it is not tied to a version before it's completed in a releasable manner. (R&D Director)

The sprint content may be determined by the business, but it is bound to time. The estimations for the work load are done and sprint allocations are based on the estimations. Some items may take longer than estimated and the sprint allocation needs to be reviewed whether the item is released in the next version or with less functionalities in the current sprint ([Case company] 2020: Data queries).

TFS items' parental tickets Features used to be divided by the sections of the big system. Features are now divided by functionalities defined by the business and sales. Features combined and allocated quarterly provide the Roadmap for the long term planning. Each feature has multiple TFS items which described a smaller part of the functionality that can be assigned to a person. Estimations and actual working and testing hours are now divided by Tasks linked to each TFS item. TFS provides resource allocation when the hours are reported by tasks ([Case company] 2020: Capacity queries). These relations are illustrated in the following Figure 11.



Figure 11. Relations between Roadmap, Feature, TFS item and Tasks.

Figure 11 has the same color coding for responsibilities as the previous figures. Business plans the Roadmap by different Features that describe a wanted functionality. Features are broken down into TFS items for development and configuration. Tasks are linked to TFS items to report the estimated and actual hours of implementation and testing ([Case company] 2020: Capacity queries).

3.5 Analysis and Key Findings of the Current State of the Product Development Process

This section analyzes the changes made to the Product development process and the development needs of the process after the corporate changes, and lastly pulls together which development needs are selected as focus areas of this thesis.

3.5.1 Analysis of the Changes Made to the Product Development Process

As mentioned earlier, there were corporate changes initiated to the Product Development Process during this study. The major changes concerned Product Development Operations and related processes, and they are analyzed below.

First, the sprint length was shortened from one month to two weeks. This also initiated practical changes to sprint activities; sprint planning cycle, handover meetings and the lengths for certain recurrent activities within all teams. These were elaborated in detail in the previous section.

Second, the sprint planning was not only scheduled to fit the new sprint length, but also business and sales took more responsibility in defining the sprint content. Previously, as the R&D Director and CTO mainly planned the sprint content, the input will now come from business and sales, which will be specified and investigated from the technical and resource management point of view. Finally, in co-operation the sprint content will be agreed based on input from business, amount of work and resources available.

Third, in order to complete a new functionality within one sprint, all teams were aligned to work with the same version in the same sprint ([Case company] 2020: Internal documentation of working items included in the release). This has an impact on QA team's work to not to test the development items before the configurations are implemented. In other way around, the Solutions team configures on a code that has not been tested by the QA team. In addition, to provide more time for testing the release is delivered twice within two weeks.

Fourth, the requested changes to the system are now sent for the department manager in order to have an extra validation step. Another validation was added when the manager forwards the valid request to the Product Issue or Roadmap mailbox, it needs to get an Approved tag from the CTO before the TFS item is created ([Sender] 2020: Received email in Roadmap mailbox, [Sender] 2020: Received email in Product Issue mailbox). This reduces the amount of TFS items that are reviewed in sprint planning.

Fifth, due to TFS functionalities, the work amount estimations on TFS items cannot be tracked in the planning phase. Tasks are now linked to the TFS items which include the estimations, and the work load is visible in the analytics for each employee ([Case company] 2020: Capacity queries)

The final major change is the retrospective meeting and analysis for each sprint. Discussion topics are; what went well in the sprint, what made the sprint difficult to achieve and what could be improved ([Case company] 2020: Memos).

3.5.2 Analysis of the Development Needs to the Product Development Process after Corporate Changes

The corporate initiated changes developed the process to some extent. This section introduces identified development needs in the product development process after the changes.

The sprint content has been more focused into specific areas of the system with the new process, but the TFS items are still lacking the specification in detail. It would be preferable to have the information how the functionality is expected to work and what is not allowed to do, as the retrospective documentation points out.

Solutions team has pointed out according to retrospective documentation, that there is no time allocated for code review. Thus, code review is not done in detail to point out all possible issues. By reserving specific time for review there would be less issues for QA to point out ([Case company] 2020: Memos).

By doing the same version in same sprint in all times, relates to an issue of waiting other part, either code or configuration, to complete. Issue of having these empty days has risen to discussion in multiple scrum and retrospective meetings by Solutions team and QA team ([Case company] 2020: Memos).

The new process of sending change requests to managers to forward those has led to a delay. The managers don't have a sufficient time to go through all requests and those are not getting forwarded as the QA Engineer has raised in scrum meetings.

Fourth, the retrospective meetings have pointed out multiple issues and points that could be improved, but the participants of the retrospective meetings are not able to effect on the issues which leads to having same points each time, as discussed in the retrospective meeting ([Case company] 2020: Memos).

In addition, there has been discussion about the timing of the retrospective meeting. The process is missing a part where all of the teams would be ready with the release and not working on the next one. Once ECM, Java and Solutions teams are ready with the release, the QA has lots of work. These testing days have in practice shifted to be the first days of the next release for other teams and by the time the retrospective is held, three out of four teams are working on the coming release ([Case company] 2020: Memos). Preferably a common day for all teams with no active release work has been pointed out in the retrospective meetings.

3.5.3 Analysis and Comparison of the Development Needs to the Product Development Process before and after Corporate Changes

Identified development needs prior to the corporate change have a wide scale from detailed individual items to corporate wide communications. Most importantly, the development process was recognized to take too much time from end-to-end. This was act on with the new shortened sprints.

Secondly, prior to the changes, it was recognized that upgrading the configurations in customer environments was too heavy work to do manually. Fortunately, the solution was already in the planning phase and the iterative development and testing is carried out.

Thirdly, during the interviews it was identified that no other person than R&D Director was capable of answering or knew where and how the TFS items come from. This relates to the identified lack of communication throughout the teams, departments and the organization. Both of these are handled as focus areas of this study.

Additionally, prior to the changes, the sprint planning covered TFS items regarding individual defects and fixes, a big picture was missing. Corporate initiated changes added the responsibility of the business to give the input for the sprint planning so the direction would be aligned. However, issues with quality and the vague requirements and specifications still remains regardless of the changes to the process.

From the multiple ticket management related issues some were defined as irrelevant in terms of the scope of this thesis. These were related to the missing business or technical TFS items and to the TFS item status lacking behind. The double testing of the related

development and configuration items has been aligned with the new process and all of the related items will be tested once all of them are ready. However, one identified issue will be handled as a focus area of this study and it is related to the related TFS items in different teams and the scheduling, since this has been seen as an issue after the changes in a form of empty days.

Lastly, prior to the changes, it was identified that QA team's testing environments are not adequate to present customer environments. The initiative from QA team was productive and the QA team is working on establishing one big environment. This adds value to the semi-automatic upgrade testing for bringing the customer environments up-to-date as well as saves money and time from the QA activities.

Some of the identified issues were at least affected by or even fixed with the corporate initiated changes as discussed above. New identified development needs after the corporate initiated changes are mainly regarding the process structure and timing.

Firstly, for a ten working days period, on working step - code review, has no allocated time to execute for Solutions team. By reserving specific time for review it would benefit the QA team to have less issues to point out. This is handled as a focus area of this study.

Secondly, the managers do not have a dedicated time for reviewing the product issues and change requests sent to them. This delays the process and lowers the level of visibility. This issue is out of the scope of this thesis.

Lastly, the timing of the retrospective meetings has raised up an issue of not having a common time for all teams when the sprint is finished and the next one has not yet started. This is handled as a focus area of this study. And so is the identified issue of participants of the retrospective meeting not being able to effect on the risen issues.

The focus areas of this study are discussed and further elaborated in the next section. The following Table 2 lists the identified development needs that were either fixed by the corporate changes or were out of the scope of this study.

Table 2. Development needs that were fixed by the corporate changes or otherwise ruled out of scope of this study (based on the current state analysis).

Area	Situation before the corporate changes	Situation after the corporate changes	Priority
Process	It takes several months of a change requisition to get implemented into customer environment.	Urgent change request can go through the process within a month.	Fixed
Content	Sprint planning focuses on individual defects and fixes.	The sprint content more focused into specific areas of the system with the sprint planning input from business.	Fixed
Process	Some items are tested twice in QA, first the development item is tested and secondly related configuration item is tested.	Development items linked to configuration items are only tested when the configuration is done.	Fixed
Testing	The QA team doesn't have a large environment with loads of data and users to present the customers' environments.	The QA team is working on establishing one big environment.	Fixed
Consultancy	Customers have very old versions due to manual upgrade of configurations.	Solution for semi-automatic upgrade is under iterative development and testing.	Fixed
Content	A technical ticket or a business ticket is missing. (Java team)	A technical ticket or a business ticket is missing. (Java team)	Out of Scope
Process	A step between QA Signoff and Active would be needed to indicate the item is waiting to be worked on. (Java team)	A step between QA Signoff and Active would be needed to indicate the item is waiting to be worked on. (Java team)	Out of Scope
Process	When the sprint is closing, some TFS items might be still in process. (QA team)	When the sprint is closing, some TFS items might be still in process. (QA team)	Out of Scope
Process	N/A	The managers don't have a dedicated time to go through all requests and those are not getting forwarded.	Out of Scope

As seen in Table 2, multiple identified development needs were covered by the corporate initiated changes, and other development needs were considered as out of scope of this thesis.

3.5.4 Selected Focus Areas of the Development Needs of the Product Development Process

The selected focus areas are presented in Table 3 below.

Table 3. Development needs considered as focus areas of this study (based on the current state analysis).

Area	Situation before the corporate changes	Situation after the corporate changes	Priority
Communication	The lack of communication within teams, departments and the whole organization	The lack of communication within teams, departments and the whole organization	Focus area
Communication	Employees have no visibility to the planning process	Employees have no visibility to the planning process	Focus area
Requirements	Issues with quality, too vague requirements and specification	The TFS items are still lacking the specification in detail	Focus area
Process	The needed development in the ECM side is in planning phase when Java team would already need it.	Issue of having empty days	Focus area
Process	N/A	There is no time allocated for code review (Solution team)	Focus area
Process	N/A	The timing of the retrospective: The process is missing a part where all of the teams would be ready with the release and not working on the next one.	Focus area
Process	N/A	The participants of the retrospective meetings are not able to effect on the issues.	Focus area

As seen in Table 3, selected focus areas are divided into three categories; culture, content and process. Communication development needs are lack of communication and lack of visibility to the entire process. Requirements development need was identified before the corporate changes and it remains after the changes; the requirement and specification of the TFS items are too vague. Process development needs are related to the common development work division and scheduling, allocation of time for code review and retrospective meetings, and forwarding the identified issues in retrospective meetings.

4 Existing Knowledge and Selected Best Practice on Improving Agile Processes

This section select best available knowledge for improving Agile processes, Scrum framework, process development methods for Agile and communication, requirements and specification in Agile process. These topics have been selected based on this research's focus areas identified in the Section 3.

4.1 Agile Process Structure and Definition

As Shore and Warden (2021) describe, Agile is philosophy, it is not something you can do; it is three things: the name, the values and the principles. The Agile Manifesto and Principles are presented in the next sub-sections as well as Scrum framework for Agile processes.

4.1.1 Agile Manifesto and Agile Principles

According to Shore and Warden (2021), the Agile Manifesto's includes the following key points:

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

The idea is to acknowledge the items on the right have value, but the items on the left are more valuable. The Agile Principles are shown in Figure 12 below.

Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity, the art of maximizing the amount of work not done, is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Figure 12. Agile Principles (Shore and Warden 2021: The Agile Manifesto).

As Figure 12 shows, the principles behind the Agile Manifesto highlights customer satisfaction with early and continuous delivery of a valuable software. This can be achieved by harnessing change of the requirements, frequent deliveries, collaboration of the developers and the business people, gathering, supporting and trusting the motivated individuals, emphasizing the face-to-face communication, measuring success by the software workability, promoting sustainable development, keeping up high quality and simplicity, empowering self-organizing teams and finally, frequent team reflection on effectivity. (Shore & Warden 2021: The Agile Manifesto.)

Other good sources on Agile are *Clean Agile: Back to Basics* by Robert Martin (2019) or *Agile for Everybody* by Matt LeMay (2018).

4.1.2 Process Definitions in Agile

While Agility embraces individuals and interactions over processes and tools, and responding to change over following a plan according to the Agile Manifesto, processes tools and plans are still needed and many sources provide benefits for having defined-process in agile teams. Having good processes, which add consistency and level expectations, can support the Agile principles and help teams deliver valuable software faster. Teams work faster when everyone is on the same page. (Brizard 2015: 16 Some Process Required.)

Brizard provides multiple examples how some practices have worked in his experience. One coherent example for this research was a process for code reviews. When there are multiple developers in the same codebase, it is critical that all the code adhere to the architectural and quality guidelines. Despite the risk of a bottleneck, dedicated code reviewers were appointed, who is responsible for completing each code review. This worked very well in the team. (Brizard 2015: 16 Some Process Required.)

Aiello and Sachs (2016) describe that well-defined process provide guidance and clarity on tasks, what needs to get done and who is responsible for completing each task. With a guidance and structure from defined development process, costly mistakes can be avoided and the software can be delivered on time and within budget. (Aiello and Sachs 2016: 3. Agile Application Lifecycle Management.)

Scrum is not a standardized process, instead Scrum is a framework for organizing and managing work – adding consistency, levelling expectations, providing guidance, clarity and structure. (Rubin 2012: 2. Scrum Framework.) Scrum, framework for Agile processes, is described in Section 4.1.3 below.

4.1.3 Scrum

According to Rubin (2012), Scrum is an agile approach for developing innovative products and services. Diligently applied Scrum benefits the organization with fast results, improved return on investment, reduced costs and higher customer satisfaction. (Rubin 2012: Can Scrum help you?)

Scrum framework including the roles, activities and artifacts are illustrated in Figure 13.

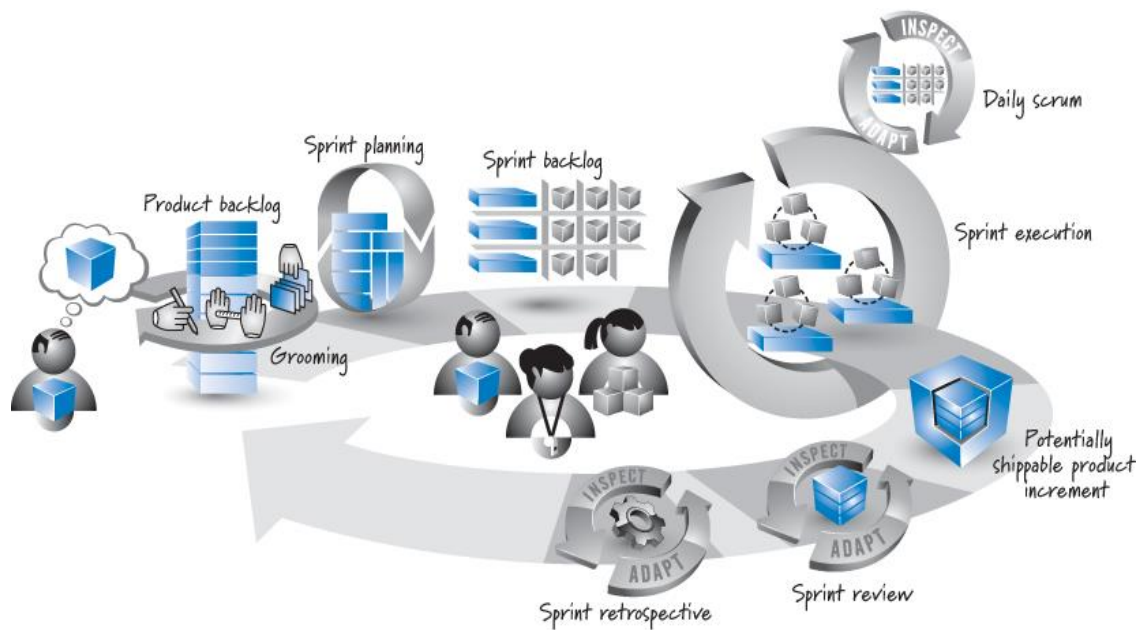


Figure 13. Scrum framework (Rubin 2012: Scrum activities and artifacts).

As seen in Figure 13, Scrum framework requires three roles. First, the product owner is the single authority with responsibility of which features and functionality to build and in which order. Second, the ScrumMaster acts as a coach to excel the productivity and helps everyone involved understand and embrace the Scrum values, principles and practices. Third, the development team consists of diverse, cross-functional people, who are responsible for designing, building and testing the product. (Rubin 2012: Scrum activities and artifacts.)

In Figure 13, the process starts with the product backlog, which is a constantly evolving artifact and consists of new features, changes to existing features, defects, technical improvements and so on. These items can be added, deleted and revised if and when business conditions change or the understanding of the items and product grow. The product backlog is groomed frequently by sizing and prioritizing the items. In Scrum, the most valuable work gets done first.

According to Rubin (2012), the work is done in iterations, called sprints, which are time-boxed with fixed start and end date, and all sprints should be of the same duration up to a calendar month. In each sprint, the work completed should create something of tangible value to the customer or user. Once a sprint is completed, the new one starts immediately. (Rubin 2012: Scrum activities and artifacts.)

Sprint planning is the first activity of every sprint. Product backlog as a basis the product owner, development team and ScrumMaster determine the items to build in the next sprint and thus conduct the sprint planning. A sprint goal is agreed on to define what the upcoming sprint is supposed to achieve. In a two-week sprint, the sprint planning typically takes four hours. In order to know which targeted features can be build, the development team breaks down each feature into a set of tasks and provides a time estimation for each task. Targeted features of the sprint and the tasks they consist of form a second backlog called the sprint backlog.

In sprint execution, the development team, guided by the ScrumMaster's coaching, performs all of the task level work necessary to get the agreed features done. Sprint execution takes up the majority of time of each sprint. A daily scrum meeting is held each day of the sprint. The agenda is to all participants very briefly describe what they've done since the last scrum and what they'll do before the next scrum and if there has been any obstacles or other issues along the way. The ideal lengths for a daily scrum is max 15 minutes, when development teams have 4-9 members. (Rubin 2012: Scrum activities and artifacts.)

Sprint results are referred as potentially shippable product increments. This means that whatever the Scrum team agreed to do, follows the agreed-upon definition of done. For example, a definition of done should comprehend a complete part of product functionality that is designed, built, integrated, tested and documented. Whether the potentially shippable product increments are truly shipped for internal or external use, is a business decision.

Sprint review's goal is to inspect and adapt the product that is being built. Participants include all interested parties; the Scrum team, stakeholders, sponsors, customers, and interested members of other teams. The focus is on reviewing the completed features in the context of overall system development. All participants have an opportunity to help guide the system development to ensure the solution created is the most business-appropriate.

Last activity in a sprint before starting the next one is sprint retrospective. The focus, instead of the product, is to inspect and adapt the scrum process and associated technical practices (Rubin 2012: Scrum activities and artifacts). Participants should cover the whole Scrum team; the development team, the ScrumMaster and the product owner.

While having the product owner might be a risk for complete honesty and revealing difficult issues, the product owner is a critical part of the Scrum process. If an issue rises from product owners activities, it would be difficult for the Scrum team to brainstorm potential process improvements if the product owner does not attend the retrospective meeting (Rubin 2012: 22. Sprint Retrospective).

According to Cohn (2009), the ideal Scrum team size is five to nine individuals and too large Scrum teams result in too much communication overhead. With an ideal size of a team, the advantages are less social loafing, constructive interaction, less time spent coordinating, no fading into background, more satisfied members and less harmful over-specialization. Additionally, small team members are more committed to their team, more aware of the goals, better acquainted with other team member's personalities, work roles and communication styles, and reported higher levels of rapport (Cohn 2009: 10. Team Structure).

If the organization does not have only one small system to develop by a single Scrum team a variations are needed. For more complex cases there are two solutions for team divisions when one large system is developed. Multiple Scrum teams can be divided as feature teams, when a cross-functional and cross-component team is responsible for end-customer features from the product backlog to the completion. Another division for teams is component teams, when a team is responsible for a subsystem that can be used to create only part of an end-customer feature. Scrum favors feature teams, but many organizations prefer component teams. Figure 14 illustrates component teams with two products and relations to all backlogs (Rubin 2012: Feature teams versus component teams.)

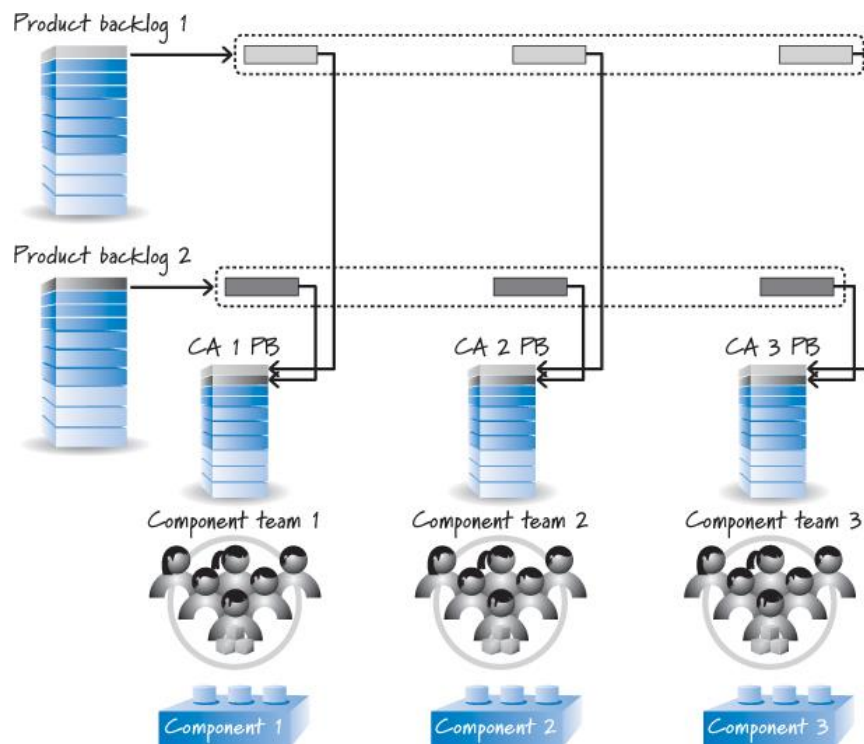


Figure 14. Two products and multiple component teams (Rubin 2012: Feature teams versus component teams).

Managing the relations between product backlogs, multiple sprint backlogs and prioritizations gets complicated when there are multiple products. Using component teams this way, increases the probability that a feature won't get finished, due to multiple points of failure, each component team, instead of a single feature team. In order to maintain the integrity of the individual component areas, which may be a concern with feature teams, the component teams and feature teams can coexist like in Figure 15 below.

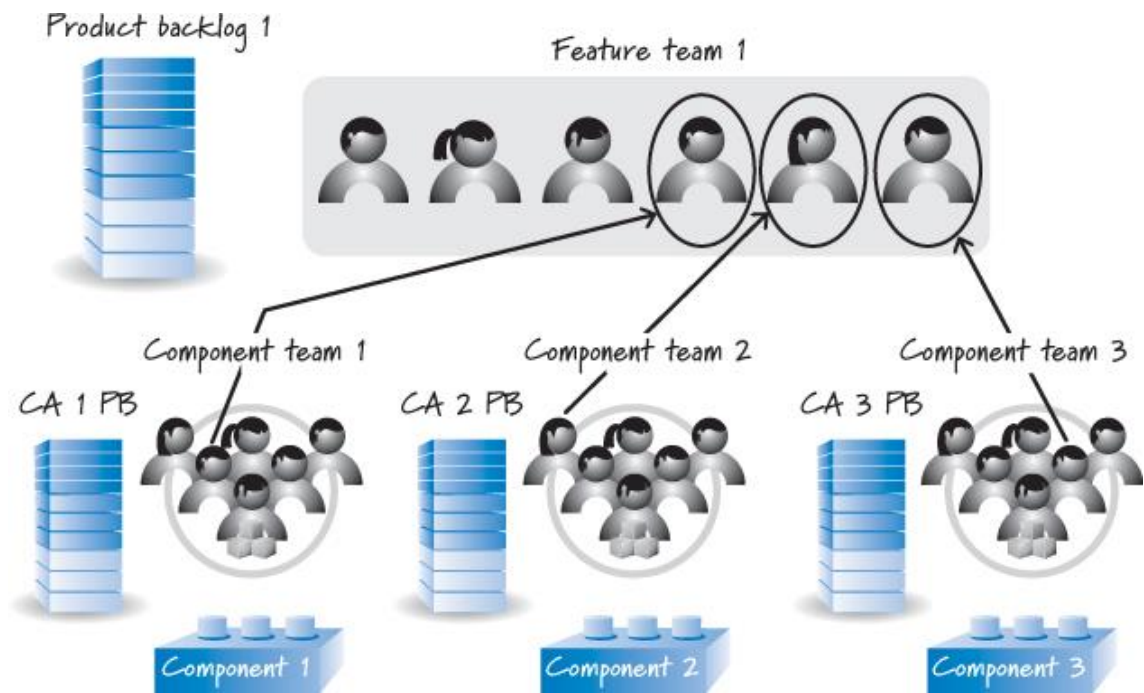


Figure 15. Combined feature team and component teams (Rubin 2012: Feature teams versus component teams).

As Figure 15 illustrates, in a combined model a feature team consists of specialists from each components and the component teams still remain with own product backlog of technical oriented items (Rubin 2012: Feature teams versus component teams).

Summing up, Section 4.1 presented Agile processes and a framework for those, Scrum. Next Section 4.2 addresses how to develop and improve Agile processes.

4.2 Process Development for Agile

This section presents two different methods to improve organization and processes towards agility.

4.2.1 Agile Fluency™ Model

According to Shore and Warden (2021), the reached results with Agile depends on company's buy-in of the Agile philosophy. They have introduced 1.1.1 Agile Fluency™ Model to see improvements from Agile zone by zone and even one zone brings benefits as long as it has been fully invested in. The model is not a maturity model, since the

continuous changes and improvements never end with Agile, but it's a matter of choosing the right cost/benefit trade-offs.

According to Shore and Warden (2021), the first zone is Focusing Zone. In Focusing Zone the organization needs to buy into the Agile philosophy and in practice the Focusing teams review processes and change plans regularly in a self-organizing manner, frequently report progress by showing what has been completed and expect to be judged on their teamwork abilities as they do their own task breakdown and allocate the tasks by themselves.

To fully adapt the Focusing Zone the following investments need to take place. First, compose thriving cross-functional teams with cooperative and skilled people with suitable backgrounds. Second, form a workspace that focuses and increases productivity. Third, assign a person with business value expertise to participate in planning sessions and to be responsible for setting product direction and goals. Fourth, assign one full-time team member to be responsible for determining the details of software abilities and capabilities. Fifth, align HR policies to support teamwork, rather than individually-focused rewards. Sixth, train managers how to manage the agile teamwork. Seventh, focus on obtaining buy-in from the key stakeholders that the value will be delivered as well as the ones who teams won't be able to provide useful estimates or forecasts. Finally, coach and train team members in the Focusing practices.

According to Shore and Warden (2021), the biggest benefits to expect from investing in Focusing fluency are that the teams plan the work in terms of business value and align it to business priorities, progress is demonstrated frequently and in terms of business value, and business priorities change directs teams' work. Additionally, management knows if teams' direction is wrong and they can intervene early, teams regularly improve their effectiveness and reduce costs, and well collaborative teams reduce misunderstandings and hand-off delays. After investing in Focusing Zone, the company has aligned itself with the core Agile philosophy.

The second zone is Delivering Zone. The Delivering Zone aims to long-term success with investing in technical success. The system is designed in a way that it can absorb frequent changes with high quality code. The production lifecycle is refined so releases and operations are painless and manageable. Delivering teams are capable of delivering reliable, low-defect software whenever it's the most sensible for the business.

The investments for Delivering Zone relate to a shift in team members' skills. First, all the Focusing Zone investments need to be made. Second, provide time for the team members to learn new skills with lowered productivity. Third, integrate all technical disciplines into the team from programming to QA and Operations. Finally, provide training and mentoring in the technical practices.

According to Shore and Warden (2021), the biggest benefits to expect from investing in fluent Delivering teams are that the latest work is released at minimal risk and cost whenever the business desires, flaws in production lifecycle are discovered and fixed early and the teams are able to provide useful predictions upon request. In addition, defect rates and codebases technical debt are low, which means less time fixing bugs, cheaper and faster changes and improved job satisfaction, morale, retention and productivity.

The third zone is Optimising Zone. Optimizing teams experiment and learn with constantly optimizing their product plans to achieve the most value possible. Team understands what the market wants, what their business need and how to bridge these two.

The investments for Optimizing Zone relate to a shift in organizational structure. First, all the Focusing Zone investments need to be made. Second, assign the responsible of business value as fulltime to the team to determine how the software fits into its market and how to achieve maximum value. Third, give full responsibility over a particular product or market segment to each team as well as responsibility for the budget, plans and results. Fourth, enable and expect managers to work collaboratively across the organization in order to remove obstacles to team performance. Finally, coach managers how high-performance, self-organizing Agile teams change management work's nature.

According to Shore and Warden (2021), the biggest benefits to expect from investing in fluent Optimizing teams are that delivered products meet business objectives and market needs, progress is described with product's stand in their market and how to improve the position and negotiations are rapid and effective with mutual trust between teams and organization. In addition, cancelling or pivoting low-value projects are coordinated early with leadership and learning from market feedback customer needs are anticipated and new business opportunities are created.

The final largely speculative zone is Strengthening Zone – a possible future for Agile. Strengthening Zone is only appropriate for organizations on the edge of management theory and practice. Shortly, it comprehends distilling teams' collective insights and channelling them into organizational improvements. (Shore & Warden 2021: The Agile Fluency™ Model.)

Nurdiani et al. (2019) investigated Agile maturity models and their order of practices. Their literature survey revealed 12 Agile maturity models and the Agile Fluency Model was considered as one even if it is distinctively different from the others. (Nurdiani et al 2019:5-7.) Next Section 4.2.2 presents another Agile process development method, Agile Process Maturity.

4.2.2 Agile Process Maturity

While Warden and Shore (2021) emphasize that Agile Fluency Model is not a maturity model, Aiello and Sachs (2016) address agile process maturity as follows:

In some ways agile process maturity could be understood almost in terms of a purity measure. Agile processes that adhere closely to agile principles would, in these terms, be considered a more agile process and, obviously, processes that just embrace some agile processes would be more of a hybrid waterfall-agile process. (Aiello and Sachs 2016: 4.2 Why is agile process improvement important?)

Agile process maturity can be understood as an adherence level of the practices in terms of the Agile Manifesto and Agile Principles. Referring to the Agile Manifesto and Principles, Aiello and Sachs (2016) emphasize the priorities to be individuals and interactions, working software, customer collaboration, and responding to change. With these as basis agile process maturity assessment can be started with investigation of what in the existing practices work well, what needs to be improved, whether prioritization is based on risk and whether process improvement is inclusive, pragmatic, agile and Lean, and if the process improvement is considered as a marathon instead of a sprint. (Aiello and Sachs 2016: 4.2 Why is agile process improvement important?)

According to Aiello and Sachs 2016, Agile process maturity has been presented with ten key points. First, process practices adhere to the Agile Manifesto and Agile Principles. Second, just like any other process an agile process must be repeatable. In other words,

the process has the same results, regardless who is performing the task – with an assumption of the proper level of skills and training. Third, agile processes must be scalable, in order to have other teams successful as well. Four, agile processes must be comprehensive to the end that everyone understands what needs to be accomplished with which interdependencies and deadlines. Five, in order to everyone to understand what is being done, how their work impacts the work of others and how their own work is impacted by others' work, process practices must be transparent. Additionally, to be able to verify the completed steps, traceability is required. Six, IT governance in place provides visibility into the organizational processes and existing operation, which guides senior management to more accurate decisions based on the available information. Seven, mature agile processes can coexist with non-agile processes. Eight, mature agile processes can align and harmonize with industry standards and frameworks. Nine, mature agile processes embrace comprehensive plans but also admit when there is not enough information to specify the details of a plan, and make the decision at the “last responsible moment”. Ten, a mature agile process embraces continuous improvement at deep technical level, throughout the software and systems lifecycle, but also at a pragmatic business level (Aiello and Sachs 2016: 4.4 Understanding agile process maturity).

Nurdiani et al. (2019) gathered 12 Agile maturity models and investigated the order of practices presented, but yet the results suggest that there is no universal strategy for Agile adoption that works better than others. (Nurdiani et al 2019:2-3.) Section 4.3 below gathers the existing literature and selected best practices for communication in Agile processes.

4.3 Communication in an Agile Process

Even though teams must be able to communicate and collaborate effectively, corporate team divisions may begin to act as if they were separate companies and this creates destructive and dysfunctional “siloeed” mentality. Poor communication leads to lack of clear understanding of what needs to be accomplished and how one's work affects the other team members. Poor communication leads also to lack of trust between the most senior managers competing with each other through corporate politics, which can be seen as groups working against each other. (Aiello and Sachs 2016: 2.16 Communicating with all stakeholders.)

According to Jamillo (2016), poor internal communication impacts on engagement. Communication is the nexus that takes all the engagement work and externalizes it to employees, to prospects and to partners, in order to ensure culture is shared and understood. It is essential to communicate with employees consistently why the company exists, why the work they do matters, and where you are heading. (Jamillo 2016:143.) Moreover, according to Jamillo (2016), when we can expect certain specified structures in our lives, we can dedicate our time and energy to areas that need it. This works to communication as well. By knowing exactly what, when, where and with whom you need to routinely communicate, you no longer have to worry whether your manager has forgotten you or what is expected from you. Expected and reliable cadence in communications frees up our mental faculties, focus, and productivity for the tasks at hand. (Jamillo 2016:194.)

By having cross-functional teams, communication and collaboration across the enterprise improves significantly. It is essential to help the team understand how the systems and software development effort affects the entire organization. This is important in order to align the technology efforts with the business objectives. (Aiello and Sachs 2016: 19.1 Goals of integration across the enterprise.) The Section 4.4 below presents different views from the existing literature and selected best practices for requirements and specifications in Agile processes.

4.4 Requirements and Specifications in an Agile Process

Requirement and specification are often used as synonyms. However, according to Heath (2020), those are two different items and closely related to each other. A requirement comes from the stakeholders and is an expression of a need, wish or desire with regard to the system being built. Requirements can be provided as formal statements, business rules, examples, user stories, business processes, screen mock-ups, narratives, charts or diagrams. To implement and deliver requirements, context and scope must be given and fitted within a user workflow, which means the requirements need to be turned into specifications. A specification is a way of defining how a requirement is going to realize. To ensure the system is what the stakeholders need and want, requirements need to be analysed and mapped into proper specifications (Heath 2020: The nature of requirements and specifications).

According to Heath (2020), user stories are not accurate enough to be used as specifications. User stories are merely attributes, short descriptions of features or sections of requirements (Heath 2020: Crossing from requirements to specifications). While Cohn (2009) agrees that user stories are not meant to be a complete feature description, but rather than documenting the total specification of the feature, he stresses that the user stories serve as a starting point for a discussion of planning the total feature between Product Owner and development team. The Product Owner's promise of availability allows the teams to accept work into a sprint without having considered all details. (Cohn 2009: Shift from documents to discussion.)

The requirements need to be relevant, valid and verifiable. The issue of requirements being understood correctly, lies in recognizing which part of the requirement is truly understood and which part needs further defining. One solution is clearly defined test cases that provide much of the same information as a requirements document. Additionally, test cases tend to be easier to keep updated and current. Mistakes in requirements definition can result in serious defects, including security risks as well as increased costs and delays in project delivery. (Aiello and Sachs 2016: 2.6 Defining Requirements.)

Brizard (2015) tells with examples how development team's sprint blockages can be avoided by having criterion on User Stories before those can be brought into Sprint backlog. Teams using criterion this way have had more predictable velocity.

Another Brizard's (2015) example shows that having a process for user stories scope changes during a sprint takes the extra work's pressure off the developers, keeps the commitment on the original point and shifts the pressure on the business to write good User Stories. (Brizard 2015: 16 Some Process Required.)

Next, Section 4.5 combines the presented views from the existing literature and selected best practices for all three categories; Process, Communication and Requirements, into a conceptual framework of this thesis.

4.5 Conceptual Framework of This Thesis

This section pulls together the previously presented best practices for each selected focus areas from current state analysis in a form of a conceptual framework, which is presented in Table 4 below.

Table 4. Conceptual framework of selected best practice elements to guide the Agile processes improvements in this thesis.

Process Development for Agile Process

CSA Focus areas	Process	Communication	Requirements
1) Issue of having empty days	Temporary Feature teams (Rubin 2012: Feature teams versus component teams) Adjusting sprint timing (Rubin 2012: Scrum activities and artifacts)		
2) There is no time allocated for code review (Solution team)	Allocate time Dedicated code reviewer(s) (Brizard 2015: 16. Some Process Required)		
3) The timing of the retrospective: all of the teams would be ready with the release and not working on the next one.	Keeping the sprints time-boxed Adjust sprint timing (Rubin 2012: Scrum activities and artifacts)		
4) The participants of the retrospective meetings are not able to effect on the issues.	Product owner into retrospectives (Rubin 2012: 22. Sprint Retrospective)		
5) The lack of communication within teams, departments and the whole organization		Cross-functional Feature teams (Shore & Warden 2021: The Agile Fluency™ Model) Sprint reviews (Rubin 2012: Scrum activities and artifacts)	
6) Employees have no visibility to the planning process		Sprint Planning Sprint review (Rubin 2012: Scrum activities and artifacts)	
7) Issues with quality, too vague requirements and specification. The TFS items are still lacking the specification in detail			Criterion for getting into Sprint backlog (Brizard 2015: 16 Some Process Required) Test cases as requirement (Aiello and Sachs 2016: 2.6 Defining Requirements) Temporary Feature teams (Rubin 2012: Feature teams versus component teams)

As Table 4 shows, the selected focus areas and the solutions gathered from existing literature and best practice are divided into three categories; *Process*, *Communication* and *Requirements*.

The *Process* related issue “having empty days” could be addressed with forming temporary feature teams for development items that require work from multiple teams. Additionally, by adjusting the timing of sprint activities would affect positively on the issue of having empty days. Second, the process related focus area is “no time allocated for code review” could be simply handled with allocating the time based on estimation and additionally dedicating a person as responsible for it. Third, the issue of timing the retrospective meetings could be addressed by truly keeping the sprints time-boxed, no starting the next sprints items before the last sprint truly ends. Additionally, by adjusting the timing of sprint activities ways in which release would take place on the second last day in order to have the retrospective meeting on the last day. The last process related focus area is “retrospective meeting’s participants are not able to effect on the discussed issues” would require participation of the product owner to the retrospective meetings, at least partially.

The *Communication* category includes two selected focus areas; “lack of communication within teams, departments and the whole organization”, and “employees having no visibility to the planning process”. The communication could be addressed by having cross-functional feature teams with a responsible person from the business side, having sprint review meetings open for all interested parties, and having the development team responsible for dividing features into tasks and estimating the working hours in sprint planning.

The *Requirements* category includes an issue identified both before and after the corporate changes; too vague requirements and specification. As the two previous categories would benefit for having feature teams, would this one too. Feature teams having participants from multiple component teams could together define how the new feature could be done and ask for needed additions from all the aspects at once. Additionally, criterion for the requirements could be stated before the item gets to the sprint planning. Finally, the requirements could be presented as test cases, as it guides the developers directly to the end result. This conceptual framework is used as a basis for building the initial proposal in Section 5.

5 Building Proposal for Improvements to the Development Process of a Standardized P2P System

This section merges the results of the current state analysis and the conceptual framework towards the building of the Proposal using Data 2.

5.1 Overview of the Proposal Building Stage

This section presents the steps in the Proposal building for this study. The current state analysis identified the development needs in the Development process of a standardized P2P system, in the Product Development Operations, and in the related processes. The development needs were prioritized into focus areas and grouped into three categories; Process, Communication and Requirements. Based on the identified suggestions from existing literature, the conceptual framework was built to address the focus areas of the development needs.

Building the proposal was conducted with stakeholders related the Development Process of a Standardized P2P System in a co-creation workshop. In the workshop, first, the identified development need and possible improvements were discussed. Second, the improvement suggestions from literature merged into the conceptual framework was presented and discussed. In practice, the conceptual framework was presented to the stakeholders later, after eliciting their improvement insights, in order to get fresh ideas and not to be influenced too early by the conceptual framework. Third, all findings were pulled together to form an improvement proposal.

5.2 Process Related Development Needs

The development needs related to the *Process* were identified, prioritized and grouped in the current state analysis. Process related development needs are the issue of having empty days, no time allocated for code review, timing of the retrospective meetings, and the retrospective participants are not able to effect on the issues encountered. The conceptual framework was presented after the identified development needs and possible improvements were discussed.

The issue of having empty days was elaborated and it was discussed with the Senior Software Engineer and the Head of Solutions team, that sprint planning within teams has

been introduced in Solutions team and in Java team to some extent. The Head of Solutions team suggested that a possible solution would be to practise these to the full extent and focus on dividing the workload in team sprint planning sessions. The literature suggested improvements, temporary feature teams and adjusting the sprint planning, got positive but realistic feedback. Temporary feature teams consisting of participants from Java, Solutions and QA teams, would be reasonable for bigger items to develop, according to the Head of Solutions team. The QA Engineer and the Senior Software Engineer saw temporary feature teams as a good improvement, creating better commitment to the developed items.

The issue of no time allocated for code review in Solutions team was discussed.

Either the time for code review needs to be added into the working hours' estimation and reported accordingly, or the length of active configuration time in a sprint needs to be shortened in order to have time for code reviews. (The Head of Solutions team)

Allocating the time and dedicating a certain person for code reviewing, as literature suggests, would have its benefits in streamlining the similar functionalities in the system, but would also mean less time for configurations for that person, according to the Head of Solutions team.

The timing of the retrospective meeting did not raise any suggestions for improvements amongst stakeholders. The conceptual framework suggests to keep the sprint time-boxed and to adjust the sprint timing in order to have the retrospective within the current sprint, not in the beginning of the next one. This was perceived as difficult, but not impossible by all the stakeholders.

The last development need related to the process is the retrospective participants are not able to effect on the issues discovered in the meetings. The QA Engineer stated that at least feedback to the retrospective documentation would be needed, in order to not to discuss the same issues in the meetings. The literature suggestion to have the Product Owner in the retrospective meetings, was more likely seen to be possible if the Product Owner would attend at least to some part of the meeting.

5.3 Communication Related Development Needs

The current state analysis pointed out that development needs related to *Communication* include lack of communication within teams, departments and the organization, and the development operations' employees having no visibility to the planning process. Once again, the development needs were first freely discussed and brainstormed and later the literature suggestions merged into the conceptual framework were presented and the feedback was received.

Lack of communication within teams, departments and organization was discussed.

This could be addressed by all teams, Java, Solutions, QA and ECM, having their own daily scrum meetings. Thus, the communication within teams would be regular and only essential issues and questions were handled in the scrum meeting common for all teams making it also shorter in time. (The Head of Solutions team)

In addition, the QA Engineer pointed out that upcoming topics in R&D could be informed more frequently. The framework's suggestions of having cross-functional feature teams with a business responsible person and sprint review meetings open for all interested parties, were not totally dismissed. Limited resources in all departments and low attendance rate in sprint review meetings raised some concerns with stakeholders.

The issue of employees' having no visibility to the planning process raised some thoughts that the company's internal communication of the current and upcoming topics and news could be more formalized and frequent. However, this is out of scope of the research objective. The literature suggestions were to give more responsibility to teams in sprint planning to divide the features into tasks and providing the estimations, and to have sprint review meetings open for all interested parties. The sprint review meetings were agreed to widen the employees' visibility to other departments and functions in the company, only if there were active participants. The Head of Solutions team had some concerns on the sprint planning suggestion of how much time would it take.

5.4 Requirements Related Development Needs

The current state analysis identified a development need related to *Requirements*, which was conclusive before and after the corporate driven changes – too vague requirements and specifications.

The open discussion with stakeholders resulted in three improvement suggestions.

There are multiple great external trainings about how to form quality requirements that could be provided for the people responsible of the requirements. (The QA Engineer)

In addition, the Head of Solutions team suggested that requirements could follow the size of the item, the bigger the item – the more detailed requirements should be. It was also mutually discussed, with the Senior Software Engineer, the QA Engineer and the Head of Solutions team that the sprint planning meetings, with the agenda of dividing the workload, should be properly in place. Thus, the team members could come prepared to the meetings by familiarizing themselves with the upcoming items, and the detailed questions of the requirements could be gathered at a very early stage.

The improvement suggestions of the conceptual framework got positive feedback from the stakeholders. A criterion for the item's requirements before it is collected into the sprint backlog was seen as a good point and it could be combined to the suggested size and requirement details relation. Writing the requirements in a form of test cases was also considered as a good development, since the test cases are a common language to all teams and the goal would be clearer to all parties. Lastly, the temporary feature teams were commonly seen increasing the commitment and the possibility to review and assess the new item together from the very beginning would make the requirements clear for all teams.

5.5 Summary of Inputs for the Initial Proposal

Table 5 below summarizes the previously discussed findings of Data 2, from the Proposal Building workshop, and the improvement suggestions from the existing literature and best practice merged into the conceptual framework and development needs identified in the current state analysis as a basis.

Table 5. Key stakeholder suggestions (findings of Data 2) for Proposal building in relation to findings from the CSA (Data 1) and the conceptual framework.

Process Development for Agile Process			
CSA Focus areas	Process	Communication	Requirements
1) Issue of having empty days	Temporary Feature teams (Rubin 2012: Feature teams versus component teams) Adjusting sprint timing (Rubin 2012: Scrum activities and artifacts) Sprint Planning within teams (Data 2) Dividing the workload (Data 2)		
2) There is no time allocated for code review (Solution team)	Allocate time Dedicated code reviewer(s) (Brizard 2015: 16. Some Process Required) Including code review time in estimated work hours or shorten the sprint length (Data 2)		
3) The timing of the retrospective: all of the teams would be ready with the release and not working on the next one.	Keeping the sprints time-boxed Adjust sprint timing (Rubin 2012: Scrum activities and artifacts)		
4) The participants of the retrospective meetings are not able to effect on the issues.	Product owner into retrospectives (Rubin 2012: 22. Sprint Retrospective) Feedback for the retrospective documentation (Data 2)		
5) The lack of communication within teams, departments and the whole organization		Cross-functional Feature teams (Shore & Warden 2021: The Agile Fluency™ Model) Sprint reviews (Rubin 2012: Scrum activities and artifacts) Own Daily Scrum to all teams (Data 2)	
6) Employees have no visibility to the planning process		Sprint Planning Sprint review (Rubin 2012: Scrum activities and artifacts)	
7) Issues with quality, too vague requirements and specification. The TFS items are still lacking the specification in detail			Criterion for getting into Sprint backlog (Brizard 2015: 16 Some Process Required) Test cases as requirement (Aiello and Sachs 2016: 2.6 Defining Requirements) Temporary Feature teams (Rubin 2012: Feature teams versus component teams) Provide training for people responsible of requirements (Data 2) Size classification based requirement details (Data 2) Sprint Planning within teams (Data 2)

Table 5 above shows the inputs for the proposal.

5.6 Initial Proposal for Improvements to the Development process of a Standardized P2P system

Proposed improvements to the Development process of a standardized P2P system, based on the development needs identified in current state analysis, gathered existing literature and selected best practices and Data 2 collection on building the initial proposal, are presented in Table 6 below.

Table 6. Initial Proposal of this thesis.

Proposal	Improvements	Issue solved
Temporary Feature teams	Commitment for the whole lifecycle of the feature, technical wise and process wise. All on the same page from design to release. Less fixes needed.	1) Issue of empty days 7) Too vague requirements
Cross-functional Feature teams	More efficient process in terms of issue encountering, requirement specification and changes	5) The lack of communication within teams/departments/organization 7) Too vague requirements
Keeping the sprints time-boxed	Less cumulative work load in testing, predictable schedule for depending departments	3) Timing of the retrospective
Adjusting Sprint timing	Working towards a common goal, all team members get to participate in all team actions	1) Issue of empty days 3) Timing of the retrospective
Sprint Reviews	More information about upcoming topics in R&D, in the company and in the customer surface	5) The lack of communication within teams/departments/organization 6) No visibility to planning process
Sprint Planning	More targeted questions about requirements. Higher commitment to the sprint goal agreed with business with ability to influence	6) No visibility to planning process 7) Too vague requirements
Daily Scrum in all teams	Shortened Scrum for all teams, only relevant collaboration required issues are handled	1) Issue of empty days 5) The lack of communication within teams/departments/organization
Allocate time for code review	Improved quality on each item	2) No time for code review
Dedicated code reviewer(s)	Improved quality in terms of uniformity	2) No time for code review
Product Owner into retrospectives	Continuous process improvement	4) Retrospective participants are not able to effect on the issues
Criterion for getting into Sprint backlog – based on size classification	Only items specified enough are handled in sprint planning	7) Too vague requirements
Test cases as requirement	The goal of the new functionality is more clear	7) Too vague requirements
Provide training of requirements	Higher quality of requirements	7) Too vague requirements

As Table 6 lists, the proposed improvements include Temporary Feature teams, Cross-functional Feature teams, Time-boxed Sprints, Adjusting Sprint timing, Sprint reviews, Sprint planning, Daily Scrum for all teams, Allocating time for code review, Dedicating a

person to code review, Product owner participating in retrospective meetings, Criterion for requirements, Using test cases as requirements, Providing training for creating the requirements.

Temporary Feature teams would consist of one person from Java team, one from Solutions team and one from QA team. Additionally if the feature is so big that it requires changes to ECM then one person from ECM team would be required too. All team members would participate in any planning session or issue solving meeting. If this is not possible due to overlapping schedules, all team members would be at least informed about changes and what caused them. When dedicated employees from each team, that the feature requires changes to, would be evaluating the requirements and working together, the commitment for the whole lifecycle of the feature is higher, technical wise and process wise. Additionally, since all employees are on the same page from Designing to Release, the process is more efficient, knowledge transfers are easier or even unnecessary, and there will be less fixes needed.

Cross-functional Feature teams would practically mean the same thing as the Temporary feature teams with an addition of a team member with the business value expertise and responsibility. The team member with business expertise would participate in all planning sessions and issue encountering meetings and would be able to give immediate answers e.g. how to solve issues and be responsible for selected solutions. Alternatively, if needed, this person would present the suggested solutions for the Product owner to decide. With a team member able to decide on encountered issues and responsible for the solutions, the process would be more efficient in terms of issue encountering and requirement changes and specifications.

Time-boxed Sprints would mean that the start and end dates of the sprint do not change. The amount of work would be estimated correctly to the sprint length in order to release the new version within the sprint and not starting the next sprint work items before it truly starts. By keeping to the agreed start and end dates in all teams, there would be less cumulative workload for QA team and the release schedule would be predictable for other departments depending on the new release.

Sprint Timing, Sprint Review and Sprint Planning are closely connected with each other. The Sprint length would remain as two weeks, but the timing within the Sprint would be adjusted in a way that the version release would be ready by the end of Day 9, so all

teams would have the possibility to participate in *Sprint Review* and Retrospective meeting within the current Sprint. The new Sprint would always start with *Sprint Planning* and the tasks in between would be adjusted accordingly, see Figure 16 below.

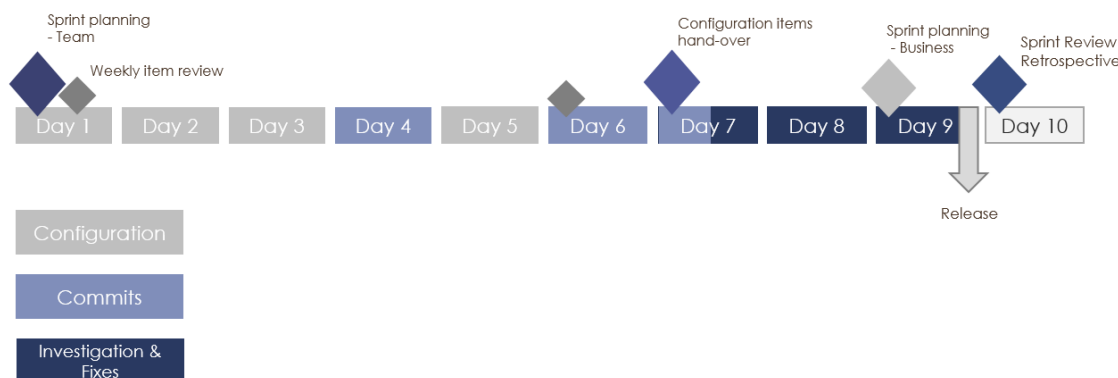


Figure 16. Adjusting Sprint timing.

Sprint review meeting would be opened for all interested stakeholders to participate. With adjusted sprint timing all teams would have the possibility to attend. In Sprint review meetings visibility and communication would be opened between teams and departments. Current and upcoming topics in R&D, in the company and in the customer surface could be easily informed to teams.

Sprint planning would start the new Sprint. It would continue to have the initial input from business side, but breaking the wanted features down to tasks and estimating the required work hours would be done by the teams. Additionally, the team would be able to divide the workload evenly between team members and by preparing beforehand by familiarizing to the wanted features, more targeted questions regarding the requirements could be presented. With the ability to influence and by agreeing the sprint goal together the teams and business, the commitment to deliver would be higher.

Daily Scrum held separately for all teams in addition to the common Scrum for all teams. Every team would check by themselves what has been done since the last Scrum and what will be next things to work on before the next Scrum, or if there are any issues encountered. In addition teams would be able to divide the workload. This would shorten the common Scrum, since only relevant, collaboration required issues and questions would be handled.

Allocating time for code review would be done either adding the required time to each task's estimated hours and reporting accordingly or shortening the sprint length in general, e.g. from 4 to 8 hours. By having the time allocated for the code review, the review would be done more precisely and the quality would be improved on each item.

Dedicating a person to code review, the time allocation would be done for one person only. In practice, the Head of Solutions team could allocate his time more to the requirements management and to code reviews in order to make the process more efficient and the systems quality in terms of uniformity would be higher.

Product owner participating in retrospective meetings would make the meetings meaningful. The discussion would not stop on the identified issues that the participants are not able to effect on. It would be ideal if the Product owner would participate at least partly to the meeting. The minimum would be receiving written feedback on the retrospective documentation. Thus, the process improvement would be continuous.

Criterion for requirements of the TFS items would reduce the time of investigations and Sprint planning. The criterion would be more comprehensive for the bigger items.

Using test cases as requirements could be also combined with the criterion. As all teams from business, to development, to QA speak the same language when it comes to the end-user use cases, the goal of the new feature would be more easily understood, the process more efficient and the goal would be clearer.

Providing external training on creating the requirements to the people responsible of defining the requirements, would be an addition to the criterion and test cases as requirements. There are multiple companies providing training for this specific and common issue. This would make the requirements quality higher and the process more efficient.

These proposed improvements will be validated by the management team in Section 6 below.

6 Validation of the Proposal

This section reports on the results of the validation stage and points to further developments to the initial Proposal. At the end of this section, the Final proposal for Improvements to the Development Process of a Standardized P2P System is presented.

6.1 Overview of the Validation Stage

This section describes the validation results of the Initial proposal developed in Section 5. Validation was conducted as a meeting with key stakeholders providing their expert judgement for the Initial proposal. The key stakeholders were the R&D Director, the Head of Solutions team and the QA Manager. The validation meeting was conducted face-to-face, except the Head of Solutions team participated via Teams.

First, all the development needs identified in the current state analysis were discussed paying special attention to both, the items that were fixed by the corporate changes, and those items that were out of scope of this thesis, and especially to the items that were selected as focus areas. Second, the proposals for improvements were brainstormed one by one and evaluated how suitable and likely the proposal item was for the case company to implement. Finally the proposal was modified based on validation feedback and the Final proposal was built.

6.2 Feedback to the Initial Proposal for Improvements to the Development Process of a Standardized P2P System

As the list of development needs identified in the current state analysis was discussed, the R&D Director agreed with all of the items that some actions need to be taken. All proposal items were evaluated based on how suitable and likely they are for the case company to implement (on four levels of suitability; with two shades of green and two shades of red presenting). The darkest green in Table 7 below presents “*very suitable*” to be implemented and the darkest red is “*not a suitable solution*” for the case company.

Table 7. Evaluation of the Initial proposal (on four levels).

	Proposal	In practice	Issue solved
	Temporary Feature teams	One person from Java team, Solutions team and QA team, additionally from ECM team if needed. All participate in planning meetings or issue solving meetings.	1) Issue of empty days 7) Too vague requirements
	Cross-functional Feature teams	A person with business value expertise to participate in planning sessions with the development team and to be responsible for setting product direction and goals	5) The lack of communication within teams/departments/organization 7) Too vague requirements
	Keeping the sprints time-boxed	Strict start date and end date. Items from the next sprint are not started before having retrospective.	3) Timing of the retrospective
	Adjusting Sprint timing	The sprint review and retrospective on the last day of the sprint, new sprint starts with sprint planning, adjusting other tasks accordingly	1) Issue of empty days 3) Timing of the retrospective
	Sprint Reviews	All interested stakeholders have the possibility to participate to sprint review	5) The lack of communication within teams/departments/organization 6) No visibility to planning process
	Sprint Planning	Every team has their own planning session, dividing features into smaller tasks, estimating the work hours, dividing the workload, preparing beforehand	6) No visibility to planning process 7) Too vague requirements
	Daily Scrum in all teams	In addition to the Scrum for all teams, every team has their own daily scrum, checking what has been done and what is to be done, issues encountered, dividing the workload	1) Issue of empty days 5) The lack of communication within teams/departments/organization
	Allocate time for code review	Time added to estimated work hours or sprint length in terms of active work on items is shortened	2) No time for code review
	Dedicated code reviewer(s)	Dedicated person to truly inspect all solutions to fit to the current functionalities and to quality standards	2) No time for code review
	Product Owner into retrospectives	Product owner participates e.g. To the second half of the retrospective or at least gives transparent feedback on the documentation	4) Retrospective participants are not able to effect on the issues
	Criterion for getting into Sprint backlog – based on size classification	Bigger items need more specific requirements, but all requirements have some criterion before those can be considered to be worked on	7) Too vague requirements
	Test cases as requirement	End-user use case provided as a criteria	7) Too vague requirements
	Provide training of requirements	Provide external trainings for people responsible of defining the requirements	7) Too vague requirements

As seen in Table 7, three items were totally discarded and four items were evaluated as “*very suitable*” to be implemented. The rest of the items were “*suitable to some extent*” to be implemented to some extent. All of the items and their evaluations are elaborated in more detail below.

6.2.1 Feedback and Developments to Temporary Feature Teams and Cross-functional Feature Teams

The Temporary Feature Teams were presented and explained the extent they are proposed to be implemented. The R&D Director had questions related to the meetings mentioned in the description and whether those would be seen time consuming and adding bureaucracy. It was mutually discussed as naturally teams have different cycles of rush hours and down times within a sprint that the main point is to design, discuss and communicate the changes by keeping the team members in a loop. No strict attendance should be required to meetings but all should be involved or at least informed. Temporary Feature Teams were seen as adding the needed co-operation and making the total development more fluent.

The Cross-functional Feature Teams were seen as already in place with the biggest new features by the R&D Director and the Head of Solutions team. They meant that when a big new feature is under development, there is a business responsible person, the CTO, who will revise the specifications when needed. However, the current procedure does not include team members from all of the teams.

6.2.2 Feedback and Developments to Keeping the Sprints Time-boxed and Adjusting Sprint Timing

The R&D Director’s opinion about the sprints was that they are currently time-boxed and it is only some individuals who start the next sprint items during the previous sprints, when they are having empty days. The QA Manager agreed and pointed out, that if release has been prolonged it has been an exception. It was agreed that it is the goal to keep the sprints time-boxed. However, Adjusting the Sprint Timing to finalize the release on Day 9 and having the Sprint Review and Retrospective on Day 10 was not seen as a preferable change.

6.2.3 Feedback and Developments to Sprint Reviews

The validators understood the benefits of having the Sprint Review open for all interested participants to share the views between departments. The Head of Solutions team, who usually organizes the Sprint Review meetings, was concerned about multiple participants spending their time in a meeting, when there might be only one topic that interests them. More effective and suitable way for the case company would be to record the Sprint Review meeting amongst the department managers, sales and consultants and share the recording companywide so all have access to it.

6.2.4 Feedback and Developments to Sprint Planning

Sprint Planning in its full extent in the Initial Proposal got resistance from R&D Director and the Head of Solutions team who are closely involved in the planning process. It would not be possible for teams to divide features into tasks and give estimations since the business requires estimations immediately when an item is discussed in Sprint Planning. The latter part, about preparing beforehand and dividing the workload, partly taken into use in Solutions team, was seen as a good improvement that would benefit all the teams. An action point was assigned to the R&D Director; once the Sprint Planning with business is finished on Day 9, the R&D Director shares the upcoming Sprint's content via email with all the teams. Thus, the teams can familiarize to the items before having the team's Sprint Planning or Joint Design meeting where they review the items closely, what is required and how the workload could be divided evenly.

6.2.5 Feedback and Developments to Daily Scrum in All Teams

Daily Scrums in all teams were seen as a very good improvement to shorten the time of the common Scrum, all validators agreed. It was also decided that only team leaders would talk in the common Scrum telling the whole team's issues or news. In addition, the team Scrums would be scheduled to the morning and the common Scrum would start immediately after team Scrums.

6.2.6 Feedback and Developments to Allocate Time for Code Review and Dedicated code reviewer(s)

Neither of the solutions to the issue of no time for code review were seen as practical improvements. According to the R&D Director, the time for code review has been taken into account in resource planning, e.g. a team member of the Solutions team has 6 hours out of 7,5 hours per day allocated for configuring and the 1,5 hours per day are estimated for code review, commits, scrum meeting, etc. Dedicating a person for doing the code review was seen as too time consuming. Inspecting the uniformity of the configurations should be conducted in the team's Sprint Planning meeting when the items are firstly reviewed.

6.2.7 Feedback and Developments to Product Owner into Retrospectives

The Product Owner understood the issue of retrospective participants not being able to effect on the issues encountered. Whenever he has the possibility to participate the meetings he will or at least he provides feedback and his opinion on the meeting documentation. Thus, the discussion and the process improvement may continue.

6.2.8 Feedback and Developments to Criterion for getting into Sprint backlog and Test cases as requirements

The criterion for TFS items and using the test cases as requirements raised concerns about not to require too much of documentation from the business side. The QA Manager saw the benefit having test cases on requirements as this is the common language for all teams. The solutions were discussed to have the criterion light and instead of test cases, the use cases of an end-user could be described on the item.

6.2.9 Feedback to Provide Training of Requirements

Providing external training for requirements was totally discarded. The R&D Director trusts that all stakeholders are professional enough to know how to write requirements for TFS items. The feedback and developments are combined in Section 6.3 in a form of the Final Proposal.

6.3 Final Proposal for Improvements to the Development process of a Standardized P2P system

Previously discussed and elaborated feedback to the initial proposal has been gathered and combined in a form of the Final Proposal in Table 8 below.

Table 8. Final Proposal of this thesis.

Proposal	In practice	Issue solved
Temporary Feature teams	One person from Java team, Solutions team and QA team, additionally from ECM team if needed. All should be involved and at least informed about issues and changes.	1) Issue of empty days 7) Too vague requirements
Cross-functional Feature teams	A Temporary Feature team with a person with business value expertise to participate in planning sessions with the development team and to be responsible for setting product direction and goals	5) The lack of communication within teams/departments/organization 7) Too vague requirements
Keeping the sprints time-boxed	Strict start date and end date.	3) Timing of the retrospective
Sprint Reviews	All interested stakeholders have the access the recording of the sprint review	5) The lack of communication within teams/departments/organization 6) No visibility to planning process
Sprint Planning	The Sprint content shared by the R&D Director. Every team has their own planning session: familiarizing beforehand to the items, planning the uniformity of implementations & dividing the workload,	2) No time for code review 6) No visibility to planning process 7) Too vague requirements
Daily Scrum in all teams	In addition to the Scrum for all teams, every team has their own daily scrum, checking what has been done and what is to be done, issues encountered, dividing the workload. Only team leaders speak in the common Scrum.	1) Issue of empty days 5) The lack of communication within teams/departments/organization
Product Owner into retrospectives	Product owner participates e.g. To the second half of the retrospective or at least gives transparent feedback on the documentation	4) Retrospective participants are not able to effect on the issues
Criterion for getting into Sprint backlog – based on size classification	End-user use case provided as a criteria. Bigger items need more specific requirements, but all requirements have some criterion before those can be considered to be worked on	7) Too vague requirements

As seen in Table 8, the Final Proposal includes Temporary Feature teams where team members are gathered from Java, Solutions, ECM and QA team, depending on the TFS item. All team members are involved and at least informed about issues or changes to requirements, so that all aspects can be investigated. Cross-functional Feature teams

are needed for the bigger TFS items, new features, when the specification cannot be done in detail in the first place, but a business responsible person will revise it along the way. Sprint are kept time-boxed with strict end and start dates. Sprint Review meetings are recorded and all the interested stakeholders have access to the recording. Sprint planning will remain mostly as it is, some enhancements to the sprint planning, joint design, meetings within the team; as R&D Director shares the sprint content via email, all team members can familiarize themselves with the upcoming TFS items and in the joint design meetings the questions of the requirements can be gathered, the uniformity of implementations can be designed and the workload can be divided. Daily Scrums for all teams are held before the common Scrum, in order to shorten the common Scrum by only team leaders having the floor. Product Owner attends the retrospective meetings whenever possible in order to remove obstacles of issues the participants cannot effect on. Presenting Criterion for requirements of TFS items by stating use cases the end-user should be able to do, lowers the amount of questions presented about the requirements.

6.4 Actions Approved Immediately for Implementation Based on the Final Proposal

In the validation of the initial proposal, the ways of implementing the improvements were discussed.

First, as the Temporary Feature teams already exist to some extent for the bigger TFS items, new features, teams will be coordinated for smaller items, which require changes from multiple teams, as well.

Second, the Cross-functional Feature teams are almost in use for new features, as the business responsible person is assigned. However, the business responsible person usually cooperates with one person, not with the whole Temporary Feature team. If the new feature requires changes from multiple teams, all members should be cooperating.

Third, the Sprint Review meetings' recordings will be shared by the Head of Solutions team from the next sprint onwards.

Fourth, from the next Sprint Planning with business, the R&D Director shares the sprint content via email to all Product Operations employees and as a continuity the more profound Joint Design meetings in teams should be taken into use.

Fifth, Daily Scrums for all teams was announced the next day from the validation and a day after all the teams had their own Daily Scrum meetings.

Sixth, Product Owners attendance to retrospective meetings should begin with the first meeting after the validation. However, the possibility of giving feedback to the documentation if attendance is not possible might change this.

Finally, the Criterion for TFS items' requirements should be defined before implementation.

7 Conclusion

This section summarizes the thesis, discusses managerial implications, evaluates the research quality and finally, closing words are presented.

7.1 Executive Summary

The Product development operations are vital for the case company's business to address the customer needs and, most importantly, to support the sales to have continually working system with updated information security measures and with cutting edge technology and features to boost the sales. This thesis was triggered by the business challenge of having some unknown bottlenecks and hindrances in the Development Process of a Standardized P2P system. The objective of this thesis was to identify these unknown bottlenecks and hindrances and propose improvements to the Development Process of a Standardized P2P system.

The study was conducted using qualitative action research methodology. The primary method of data collection was interviews. Other data methods used included the analysis of internal documentation, meetings and workshops, and observations in internal training.

During the current state analysis, as an agile company, the case company initiated changes to Product Development Operations and processes, which added another round to this analysis. The current state analysis resulted in mapping the process before and after the corporate changes, and identification of the development needs in the Product Development Operations before and after the changes. The development needs were filtered and prioritized to meet the objective of this thesis. The development needs considered as focus areas of this study were divided into three categories; Process, Communication and Requirements. Process related development needs included, for example, issue of having empty days due to awaiting for other team to complete their related task, the timing of retrospective meetings and participants of retrospective meetings not being able to effect on the encountered issues. Communication related development needs refer, for example, to communication within and between teams and between departments. Development needs in category Requirements were solely related to too vague specifications and requirements on work items.

With the goal to find solutions to the identified development needs, the study next focused on exploring existing literature and best practice. Out of the available existing literature and best practice, the most relevant suggestions were selected to tackle the identified development needs. The search for best practice was focused on the same three categories identified as problem areas in the current state analysis; Process, Communication and Requirements. As an outcome of the literature and best practice search, the conceptual framework was built.

Based on the results of the current state analysis and guided by the conceptual framework, the initial proposal was co-created together with the project team. This initial proposal suggested improvements to the Development process of the standardized P2P system. The initial proposal had 13 proposed improvements to address the seven identified development needs. The proposal included, for example, having temporary feature teams and time-boxed sprints, having sprint planning meetings in all teams, dividing the workload and the product owner participating to the retrospective meetings to address the Process related focus areas. For Communication the proposal suggested cross-functional feature teams, sprint review meetings open for all employees and daily scrum meetings to all teams amongst other suggestions. For Requirements related focus areas the proposal included, for example, classification and criterion for requirements on the work items and using test cases as requirements.

The initial proposal was analysed and validated by three supervisors. Three proposed improvements out of the total 13 were discarded, while the remaining 10 improvements were either totally approved or approved with some modifications. Based on the feedback, the Final Proposal of Improvements to the Development Process of the Standardized P2P System was developed for practical implementation. As the initial proposal was validated, the ways of implementation were also discussed. Some of the improvements require more detailed planning before implementation, but some were introduced the next day and implemented the day after that. Additionally, it was acknowledged that in agile environment the process is observed and evaluated, and improvements are being sought continuously.

The Final Proposal consists of eight improvements, as some were merged together in development; Temporary Feature teams, Cross-functional Feature teams, Time-boxed Sprints, Sprint Reviews, Sprint Planning, Daily Scrum in all teams, Product Owner into Retrospectives and Criterion for Requirements. Implementing these improvements in

Table 10, should fix the identified development needs related to Communication, Requirements and Process, but it will require time before the results of these improvements can be fully recognized; therefore, this part is left to another researcher to analyze the outcomes and continue on the road of further process improvements. Based on the most recent feedback, the agile nature of the improvements elevated the maturity of the process agility, and improved the process structure especially by applying the Scrum framework related improvements. These improvements help the case company's business by better addressing the customer needs and, most importantly, they support the sales to have a continuously working system with cutting edge technology and features to boost the sales.

7.2 Next Steps and Recommendations toward Implementation

Some of the suggested improvements were implemented straight away after validation (see Section 6.4), but some require further planning before implementation. The main two recommendations identified as next steps relate to the following areas in the product development process:

First, in order to fully implement the suggested improvements, team leaders from all teams, Java, ECM, Solutions and QA, should be engaged to drive the changes in their teams. Thus, all cooperation related improvements, such as Temporary and Cross-functional Feature teams, Sprint Planning and Time-boxed Sprints, would truly be implemented.

Second, the criteria for requirements and using use cases as requirements needs further planning and assigning the responsibilities. The bigger the work item is, the more detailed specification is required, but to what extent? What does each criterion consist of? How many use cases should be presented? These questions require clear answers for all the work items, which could mean creating a framework for establishing the requirements, which in turn could be a subject for an entire new thesis.

7.3 Thesis Evaluation

The thesis objective is to propose improvements to the Development Process of a Standardized P2P system. However, due to high level of dependency to other processes

within Product Development Operations, the proposed improvements affect the other teams and processes as well. The biggest influence to this study was the agile environment and continuous changes and improvements to processes in Product Development Operations. Also, the agile nature manifested in multiple continuous changes that happened over the course of this study. First of all, the case company initiated changes at the moment of conducting the current state analysis, and then smaller changes or improvements were emerging now and then while conducting this thesis; so the subject of this study reminded a moving train at times. The current state analysis was stopped at some point, while changes continued, and while building the initial proposal and its validation, another round of new changes and improvements was introduced, this time triggered by and based on the results of this study. Thus, despite the agile environment and continuous changes at fast pace, the final proposal for improvements developed in this thesis was highly relevant for the case company. Moreover, it became the basis for the next round of changes. This was ensured by conducting all stages of the research process in this study in a tight schedule. The key constraints of this thesis were a siloed organization and lack of contact to other departments. By having a deeper connection to the business side in the current state analysis and in the proposal and validation stages, the outcome could be even more effective to the entire process.

As this rigorous thesis used action research and qualitative research methods, according to Saunders, Lewis and Thornhill (2016: 206-207), the research quality criteria for an action research include dependability, credibility and transferability. This thesis strengthened dependability by, first, conducting a focused current state analysis constantly checking the relation to the business problem of the case company and recording all the changes. A focused study of existing knowledge guided by the current state analysis findings resulted in the conceptual framework that informed further development steps. By following this approach, the entire research – from setting the objective to the final proposal – was comprehensively documented, grounded in empirical evidence, and high relevance was kept to the practical challenges of the case company.

The credibility was ensured by relying on extensive documentation analysis, conducting participant observations, and involving the most relevant and knowledgeable key stakeholders in development actions of this research. In addition, the research design and data collection plan were planned ahead and in detail, but also revised and corrected

according the intermediate findings of the research. Data collection triangulation was ensured via the analysis of the case company's internal documents, interviews and observations. Data was captured in multiple forms; recordings, Jamboard, files, and field notes. The bias was recognized as possible subjectivity, since the study was conducted at the researcher's place of work, for the exact process that the researcher is working in. The study started within three months when the researcher started working in the case company and scrutinizing this process. The outcome of this thesis was reviewed and revised by a management team of three supervisors.

Finally, regarding transferability, this thesis was conducted and presented in a transparent way in order to make the research process open to other teams, and thus help the results to be used also in other processes of the case company. The research design and objectives, explicit reports on the findings and interpretations, openly available existing knowledge and the conceptual framework should all help to make the results applicable for any process improvement in research settings.

7.4 Closing Words

This thesis was conducted in agile environment and despite the continuous changes, the outcome addresses the initial business problem. Even though the outcome's proposed improvements relate beyond to the thesis objective, it was essential to not to treat the identified issues isolated to one process, but rather to see the big picture and implications to other closely related teams and processes as well. The identified development needs were grouped into three categories; Process, Communication and Requirements. The solutions were sought from agile processes, agile process improvement, scrum framework, requirements management and from the project team involved in second data collection. The proposed improvements were partly implemented immediately after validation and some improvements require more planning, detailing and engaging.

References

- Aiello, B., Sachs, L. (2016). *Agile Application Lifecycle Management: Using DevOps to Drive Process Improvement*. Addison-Wesley Professional. Available from: <https://learning.oreilly.com/library/view/agile-application-lifecycle/9780132761857/> (Accessed 25 Oct 2020)
- Blaxter, L., Hughes, C. & Tight, M. (2010). *How to Research*. 4th ed. Berkshire: Open University Press. Available from: <https://ebookcentral.proquest.com/lib/metropolia-ebooks/detail.action?docID=650302> (Accessed 18 April 2020)
- Brizard, T. J. (2015). *Broken Agile*. Apress. Available from: <https://learning.oreilly.com/library/view/broken-agile-stories/9781484217450/> (Accessed 25 Oct 2020)
- [Case company]. (2020) *Retrospective meeting of the Product Development Operations*. 18th of August 2020, 3rd of September 2020, 14th of September 2020, 28th of September 2020. Memo.
- [Case company]. (2020) *Template Release Note*. 6th of March 2020, 15th of April 2020, 20th of May 2020, 12th of June 2020. Internal documentation of working items included in the release.
- [Case company]. (2020) *P2P Configuration*. 17th of August, 1st of September, 11th of September, 25th of September, 15th of October. Internal documentation of working items included in the release.
- [Case company]. (2020) *P2P Template – Sprint 10.3.1*. Available from: [case company's Microsoft Visual Studio Team Foundation Server] (Accessed 17 November 2020). Data query.
- [Case company]. (2020) *P2P Template – Sprint 10.3.2*. Available from: [case company's Microsoft Visual Studio Team Foundation Server] (Accessed 17 November 2020). Data query.

[Case company]. (2020) *P2P Template – Sprint 10.3.3*. Available from: [case company's Microsoft Visual Studio Team Foundation Server] (Accessed 17 November 2020). Data query.

[Case company]. (2020) *P2P Template – Sprint 10.3.4*. Available from: [case company's Microsoft Visual Studio Team Foundation Server] (Accessed 17 November 2020). Data query.

[Case company]. (2020) *P2P Template – Sprint 10.3.5*. Available from: [case company's Microsoft Visual Studio Team Foundation Server] (Accessed 17 November 2020). Data query.

[Case company]. (2020) *P2P Template – Sprint 10.3.6*. Available from: [case company's Microsoft Visual Studio Team Foundation Server] (Accessed 17 November 2020). Data query.

[Case company]. (2020) *P2P Template – Sprint 10.3.7*. Available from: [case company's Microsoft Visual Studio Team Foundation Server] (Accessed 17 November 2020). Data query.

[Case company]. (2020) *P2P Template – Sprint 10.3.8*. Available from: [case company's Microsoft Visual Studio Team Foundation Server] (Accessed 17 November 2020). Data query.

[Case company]. (2020) *P2P Template – Sprint 10.3.9*. Available from: [case company's Microsoft Visual Studio Team Foundation Server] (Accessed 17 November 2020). Data query.

[Case company]. (2020) *Capacity – Sprint 10.3.5*. Available from: [case company's Microsoft Visual Studio Team Foundation Server] (Accessed 17 November 2020). Capacity query.

[Case company]. (2020) *Capacity – Sprint 10.3.6*. Available from: [case company's Microsoft Visual Studio Team Foundation Server] (Accessed 17 November 2020). Capacity query.

- [Case company]. (2020) *Capacity – Sprint 10.3.7*. Available from: [case company's Microsoft Visual Studio Team Foundation Server] (Accessed 17 November 2020). Capacity query.
- [Case company]. (2020) *Capacity – Sprint 10.3.8*. Available from: [case company's Microsoft Visual Studio Team Foundation Server] (Accessed 17 November 2020). Capacity query.
- [Case company]. (2020) *Capacity – Sprint 10.3.9*. Available from: [case company's Microsoft Visual Studio Team Foundation Server] (Accessed 17 November 2020). Capacity query.
- Cohn, M. (2009). *Succeeding with Agile*. Addison-Wesley Professional. Available from: <https://learning.oreilly.com/library/view/succeeding-with-agile/9780321660534/> (Accessed 25 Oct 2020)
- Coughlan, P. and Coughlan, D. (2002). Action research for operations management. *International Journal of Operations & Production Management*. Vol. 22, Issue 2, pp. 220-240. <https://doi.org/10.1108/01443570210417515>
- Heath, F. (2020). *Managing Software Requirements the Agile Way*. Packt Publishing. Available from: <https://learning.oreilly.com/library/view/managing-software-requirements/9781800206465/> (Accessed 25 Oct 2020)
- Jaramillo, S. R. (2016). *Agile Engagement*. Wiley. Available from: <https://ebookcentral.proquest.com/lib/metropolia-ebooks/reader.action?docID=4745628> (Accessed 25 Oct 2020)
- Nurdiani, I. Börstler, J. Fricker, S. Petersen, K. Chatzipetrou, P. (2019) Understanding the Order of Agile Practice Introduction: Comparing Agile Maturity Models and Practitioners' Experience. *Journal of Systems and Software*. Vol. 156, pp. 1-20. https://www.sciencedirect.com/science/article/pii/S0164121219301207?casa_token=m2729GRhdQgAAAAA:guoq_x1eSEiSQfqzhElsYAfdt-uy7DluMtrFNjPEGgGECZYa8OGr_S8BW7OW36M9jneJkXHlBaE#sec0002 (Accessed 3 Nov 2020)
- Näslund, D. Kale, R. Paulraj, A. (2010). Action research in supply chain management – a framework for relevant and rigorous research. *Journal of Business Logistics*. Vol. 31, Issue 2, pp. 332-339.

[https://oma.metropolia.fi/delegate/download_workspace_attachment/6805423/Section%202.4.%20A%20framework%20for%20RELEVANT%20and%20RIGOROUS%20AR%20\(2010\)%20-%20ADVANCED%2C%20see%20only%20Table%202.pdf](https://oma.metropolia.fi/delegate/download_workspace_attachment/6805423/Section%202.4.%20A%20framework%20for%20RELEVANT%20and%20RIGOROUS%20AR%20(2010)%20-%20ADVANCED%2C%20see%20only%20Table%202.pdf)

Rubin, K. S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Upper Saddle River NJ: Addison-Wesley Professional. Available from: <https://learning.oreilly.com/library/view/essential-scrum-a/9780321700407/> (Accessed 25 Oct 2020)

Saunders, M., Lewis, P. & Thornhill, A. (2016). *Research methods for business students*. 7th ed. Harlow: Pearson Education.

[Sender]. (2020) *Change / Feature Request – [Replacement]*. April 2020 – October 2020. Received email in Roadmap mailbox.

[Sender]. (2020) *Log Product Issue – [Replacement]*. April 2020 – October 2020. Received email in Product Issue mailbox.

Shore, J. & Warden, S. (2021). *The Art of Agile Development*. 2nd ed. O'Reilly Media, Inc. Available from: <https://learning.oreilly.com/library/view/the-art-of/9781492080688/ch01.html> (Accessed 25 Oct 2020)

Slack, N. & Brandon-Jones, A. (2019). *Operations Management*. 9th ed. Harlow, England: Pearson.

Appendix 1: Initial interview questions to direct the interviews

1. Describe the Development Operations (with Jamboard post-its)
2. Describe the Development process of the Standardized system (with Jamboard post-its)
3. Describe the relations between Configuration and other sections in the total Development Operations
4. Describe the phases in the processes that you feel don't need improvement, if any
5. Describe the phases in the processes that you feel need improvement, if any
6. Coming back to each of the outlines five areas, which development needs do you see there, at the moment of conducting this interview? (area by area, discussed in detail)
7. Coming back to each of the outlines five areas, which possible improvement could you see there, at the moment of conducting this interview? (area by area, discussed in detail)

Appendix 2: Summary of main content points from Interview 1 (example)

Respondent 1: R&D Director

April 7th, 2020

- From the long-term, quarterly and yearly planning, the roadmap is formed
- Roadmap includes business tickets, which can be divided into development tickets
- Tickets are categorized into Defects, Enhancements and New Features
 - o We are currently focusing too much on Defects and Enhancements
 - o Instead we should focus on New Features
- Development and Configuration tickets will be tested by QA before publishing
- A major version is released once a year and a minor version is released in the end of each sprint, monthly (ECM and workflow)
- Newly created tickets go to ticket backlog, after that to joint design, which is followed by QA Signoff, next it is active and after implementation it goes to code review after code review it is committed, which determines the version/sprint, then it goes to QA for testing, reviewing and documentation.
- ECM and Javastack are different architecture, so there is an integration between
- If a request requires coding and configurations, it takes way too long to deliver the feature to the customer, e.g. the request is logged in March, Development starts coding in April, Configurations are done in May and testing in customer environment can be also done in May, so the Feature is fully delivered in June

- This results in big amount of hotfixes
- Development and Configuration should be done simultaneously
- Roadmap for the next 5 years is guided by business, customers, technology and legislation
- Sprint planning for new features is missing
- Ticket requirements are not clear
- Work estimate: sprint planning vs. release
- Technical environments; loads of data results in poor performance
 - QA should have big environments
- Coverage of testing: regression tests & tests for each ticket

Appendix 3: Summary of main content points from Interview 3 (example)

Respondent 1: QA Engineer

April 16th, 2020

- During the past year the development has shifted from Product Development Operations more to be driven by business. This means less technical innovations.
- Ticket work types:
 - Investigation: if the item can be done or not
 - Specification, what does the wanted feature require; development or configuration or both
 - QA has double work to test both tickets even though those are related
 - Configuration
 - Development
- In sprint planning the developers used to decide together
- Describing the ticket lifecycle
- There might be 200 tickets in one sprint and about 100 of those need to be documented
 - Documentation tool can be used only one person at a time
- When testing the previous sprint items end the next day testing the next sprint items start

- When a ticket comes to QA, it is first reviewed who is testing which tickets
 - Before testing a check if all the documented configurations are in place in testing environment
 - It was noticed that many times some vital configurations have been missing, so checking was added, which made the process more effective
 - After testing, the tickets are reviewed by the QA Manager
 - Tickets are reviewed if documentation to internal manual is needed
 - There is not enough time scheduled for documentation
- When tickets are closed or in documentation step and the release has been published, internal notes and customer release notes are published as well
- There have been times when testing has been done three releases late
- There are also automated tests in place in addition to manual testing
- Development environments used in configuration were discussed
 - Items are first configured to a development environment with data for initial testing
 - Next the configurations are added to the empty isolated environment manually after code review step
 - Importing the configurations from the empty environment to test environment empties all data from the testing environment

- All master data and example cases need to be imported and created
- Bad tools
 - Regression tests have separate reporting system, which is not ideal for the reporting purposes
 - TFS is difficult to use
- Communication in QA team works
 - One can ask for help and receives it (eventually)

Appendix 4: Jamboard page 2/5 from Interview 3

Ticket lifecycle; red post-its pinpoint the development needs

