

Anastasia Tuomi

Product Owner Role in City of Helsinki Agile Software Development

Helsinki Metropolia University of Applied Sciences

Master of Engineering

Information Technology

Master's Thesis

4 May 2018

PREFACE

For me, this Master's Thesis offered an experience-rich journey to the world of agile software development. I was surprised how remarkably amount of time it took me to get familiar with the Thesis subject. However, it additionally gave me many new skills that I hopefully can utilize and develop in the future. The case organization offered me a great environment to take the Thesis theory in to an action by acting as a product owner in the case study.

I express my gratitude to co-workers and specialists in the case organization for participating and supporting the case study research of this Thesis. I would like thank my instructor Ville Jääskeläinen at Metropolia University of Applied Sciences for guiding me with my Thesis. Thanks also belong for my relatives and friends who have been supporting me in my Master's studies.

Special thanks to my dearest Juha. You were there for me whenever I needed. You never lost your mind over all the time and effort I spent on working with my Thesis instead of being with my loved ones. Eternally grateful for your support; I could not have done this without you.

Helsinki, 4 May 2018
Anastasia Tuomi

Author(s) Title	Anastasia Tuomi Product Owner Role in City of Helsinki Agile Software Development
Number of Pages Date	56 pages + 5 appendices 4 May 2018
Degree	Master of Engineering
Degree Programme	Information Technology
Instructor(s)	Ville Jääskeläinen, Principal Lecturer
<p>This Master's Thesis focused to the product owner's role in City of Helsinki environment. The City Administrative Office of City of Helsinki had developed its own agile experiment and implementation method for agile software development. The method was new and there was not yet much experience of using it. The objective of this Thesis was to find out the product owner's ability to carry out the software development project by using the agile experiment and implementation method. Additionally, the organization's current ability to support the product owner in that was under inspection. The research method used in this Thesis was case study and the case study project was information system renewal project.</p> <p>After forming a conceptual framework of the subject, the case study was carried out by implementing the tasks of the product owner required in the agile experiment and implementation method. A Fit-GAP analysis was used for evaluating the product owner's ability to carry out the tasks and to find the possible improvement targets.</p> <p>The outcome of this Thesis was an evaluation whether the product owner was able to carry out the case project by using the agile experiment and implementation method. The analysis of the fits and gaps of the implemented tasks offered a valuable information about the possible improvement targets of the product owner's role in City of Helsinki environment.</p> <p>The City Administrative Office benefits from the results of the Thesis by increasing their understanding of the product owner's role in agile software development in the organization. Additionally, recommendations for improving the product owner's role in City of Helsinki agile software development environment were proposed based on the Thesis results.</p>	
Keywords	Product owner, Agile, Software Development, Scrum

Table of Contents

Preface	
Abstract	
List of Figures	
List of Abbreviations	
1 Introduction	1
1.1 Business Challenge	2
1.2 Objective and Scope	3
1.3 Outcome	4
1.4 Thesis Structure	4
2 Current State of Agile Software Development in Case Organization	5
2.1 Experiences of Agile Software Development in Case Organization	5
2.2 Agile Experiment and Implementation Method	6
2.3 Agile Software Development Environment in Case Organization	7
3 Method and Material	9
3.1 Research Plan	9
3.2 Research Model and Process	10
3.3 Data Collection and Analysis	13
4 Conceptual framework	14
4.1 Overview of Agile Software Development	14
4.2 Agile Development Methods	14
4.2.1 Scrum	15
4.2.2 Scrum of Scrums	15
4.3 Product Owner in Agile Software Development	16
4.4 GDS	17
4.5 PRINCE2	17
4.6 Lean	18
4.7 SAFe	18
4.8 Summary	19
5 Implementing Product Owner Role – CASE Huvaja	20
5.1 Project Setting	20
5.2 Experiment	24

5.2.1	Discovery	25
5.2.2	Alfa	28
5.3	Implementation	31
5.3.1	Beta	32
5.3.2	Live	37
6	Findings Analyzation and Key Results	42
7	Recommendations	45
7.1	Resourcing and Training	45
7.2	Tools and Support	47
7.3	Handbook	49
8	Discussion and Conclusions	51
8.1	Summary	51
8.2	Case Organization's Feedback	52
8.3	Validity and Reliability	53
8.4	Conclusions	53

References

Appendices

Appendix 1. Fit-GAP Analysis template

Appendix 2. Fit-GAP Analysis of tasks in case study

Appendix 3. Vision Canvas of Huvaja in discovery phase

Appendix 4. Test model

Appendix 5. Checklist for transition

List of Figures

Figure 1. Research Plan	9
Figure 2. Cyclical Research Process	11
Figure 3. Multiple-choice Fields in Fit-GAP Analysis Template	12
Figure 4. Links of Product Owner in Case Project.....	22
Figure 5. Project Backlog and Sprint Backlogs in Trello Application.....	34
Figure 6. Structure of Huvaja and Its Integrations	39
Figure 7. Fit-GAP Ratio of Case Study Tasks.....	42
Figure 8. Sources of Partial Gaps	43
Figure 9. Improvement Areas of Product Owner Support.....	45
Figure 10. Structure of Handbook.....	50

List of Abbreviations

AOK	Open Source Software Development Unit
CITO	Central IT Office
DevOps	Software Development and IT Operations
DoD	Definition of Done
DSDM	Dynamic Systems Development Method
FDD	Feature-Driven Development
GDS	Government Digital Service
KEHMET	Collection of City of Helsinki Development Methods
MVP	Minimum Viable Product
PRINCE2	Projects in Controlled Environments
SAFe	Scaled Agile Framework
SLA	Service Level Agreement
XP	Extreme Programming

1 Introduction

The aim of this Master's Thesis was to research the product owner role in agile software development in public sector organization. The research was implemented as a case study. In the case study, the case organization's new agile method was tested in the real life project. This section offers a background information of the Thesis subject and presents shortly the organization involved. In addition, the business challenge, research objective, output and the scope are presented in this section. Finally, at the end of this section the structure of the Thesis is explained.

In today's world, many software development companies has already established an agile development as part of their daily operations. More and more the companies in wide range of branches have noticed the needs for agile development in their organizations. In addition, the public sector is nowadays one actor in this field. Digitalized services should be produced in an accelerating pace. The traditional software development methods such as waterfall model are too inflexible and complicated to response to this demand. There is need for a more flexible and agile method to produce faster the outputs. However, the challenge of applying the new methods is that the organization is used to follow the traditional method and is not prepared for changing its own behavior to meet the requirements of the agile development. That may mislead to a conclusion of the agile methods inappropriateness in the organization. Investigating and evaluating of the changes and planning the implementation of the needed actions may help to apply the new method to the organization.

The organization involved in this Thesis was the City of Helsinki (hereafter referred to as case organization). The focus of the research was in the agile software development in the case organization. The case organization is divided in four division of different branches. The divisions produces the services to citizens in City of Helsinki and the needs of the business life in City of Helsinki area. [1] In addition to that, there is the central administration called City Executive Office. The City manager runs the City Executive Office and it consists six divisions. [2] In this master's Thesis, the research was carried out in one division of the City Executive Office called Information technology and Communications. The Information technology and Communications division includes the Central IT Office (CITO) of the case organization.

The case organization's software projects has usually carried out by using the traditional waterfall model. This model is well known and it supports the decision-making and acquisition processes of the case organization but it is inflexible and the implementation times of the projects are too long. The case organization has recognized the need of faster and more flexible ways to design and implement information systems. For answering to that need, the CITO has developed an agile experiment and implementation method. The method aims to support faster decision-making, more flexible project setups and implementations. [3] The main target group of the agile experiment and implementation method is the product owners, service owners and the solution architects working with the information systems in the case organization. [4]

1.1 Business Challenge

As the agile experiment and implementation method is new in the case organization, there is limited experiences of its usage. Before starting to deploy the method more widely in the case organization, its suitability need to be tested. Especially more knowledge was needed about the ability of the target group to use the method and how the development and decision making structure supports the usage of the method. Testing the method, the CITO needed a help from different divisions of the case organization to offer case projects for the testing. In the case projects, the division's software development projects should be carried out by following the agile experiment and implementation method.

The City Executive Office's own ICT department called Kanslia ICT produces the ICT services such as help desk services, infrastructure and data communications services for the employees of the City Executive Office. In addition, it offers the maintenance services for the City Executive Office's own information systems. Currently, in the maintenance of Kanslia ICT was the resource booking system that was no longer technically up to date and there was not available the necessary system vendor support. In addition, the resource booking system was available only for the limited amount of the employees of the City Executive Office. To make the use of it more effective, the resource booking system should be enabled for all the employees. However, the current solution does not support the expansion of the system.

The Kanslia ICT had identified the need to renew the resource booking system. The current system was a tailor-made solution that takes account the special needs of the different user groups. The investigation of the corresponding systems in the open market and one pilot project revealed that there is limited amount of options available and to be able to fulfill the requirements set to the system, the ready-made solutions would need a significant amount of tailoring. According to the discussions in the case organization, the Kanslia ICT saw an opportunity to carry out a renewal project and to reach to the desired outcome by following the agile experiment and implementation method. The Kanslia ICT was willing to test the method in the resource booking system renewal project.

1.2 Objective and Scope

The objective of this Thesis was to find out the product owner ability to use the agile experiment and implementation method. When researching the ability, there is two perspectives to be taken into account. First is the ability to carry out the required tasks set to the product owner in the agile experiment and implementation method. Second perspective is that how the case organization is supporting the product owner to use the agile experiment and implementation method and fulfill the requirements. The research question was outlined as following:

As a product owner is it possible to carry out the project by using the agile experiment and implementation method?

Even though the agile development includes different roles, the research scope was to evaluate only the role and responsibilities of the product owner. This Thesis offers one point of view to research the subject by using the case study approach. The results of the study bases more on the findings from practical experiments instead of theoretical evaluation of the subject.

1.3 Outcome

The outcome of the Thesis is an evaluation of the product owner's ability to carry out the case study project by following the agile experiment and implementation method. Related to that, the study includes the evaluation of the capability of the case organization to support the product owner to meet the requirements of the method. In addition, the study offers some recommendations on how the case organization can improve their ability to support the product owner. This includes, when necessary, recommendations on how to improve the agile experiment and implementation method itself.

1.4 Thesis Structure

This Thesis is divided in eight sections. The first section introduces the topic and the framework of the Thesis. After that follows the current state analysis in the section 2. The section 3 presents the methods and materials that were used in this research. The section 4 includes a literature survey of the topic. The actual implementation of the research is described in section 5 and section 6 includes analysis of the gathered data. Section 7 offers recommendations based on the data analysis. Last, there is discussion and conclusions of the Thesis in section 8.

2 Current State of Agile Software Development in Case Organization

The aim of this section is to offer an overview to the current state of software development with agile manners in the case organization. In addition, it provides an overview of the agile development maturity in the case organization environment. This section presents the purpose and structure of the agile experiment and implementation method and the used conditions more specifically. Before starting to describe the current state of the case organization, some earlier experiences of agile software development from the literature are presented.

2.1 Experiences of Agile Software Development in Case Organization

The case organization knowledge of agile software development have usually been limited to the usage of methods offered by software vendors. The method have used as a part of the implementation phase of the traditional waterfall model software development. The role and the need for competence of the case organization's project manager in these have been minor. From a technical point of view, the agile development has started to formulate around the need of supporting the service owners in their software development. For that need, they had acquired resources and competences. Later, the technical support has expanded its role for supporting the agile experiments and testing the new platforms. That technical support is nowadays known as open source software development unit (AOK). [3] The AOK is unit in the CITO. Purpose of AOK is to advance the usage of the open data, interfaces and source code in case organization's software development projects. Work in AOK is done by agile manners and they are committed to follow the agile experiment and implementation method. [5]

The agile manners of a case organization have been studied in the Bachelor Thesis of Juho Kerppola. At that time, the agile experiment and implementation method was still under development. The case study research focused to the evaluation of the usage the agile methods in the case project. As a conclusion, the researcher stated that the research offers an input to the development of an agile experiment and implementation method. [6]

Generally, experiences of agile software development in the case organization had been limited to the occasional usage of some agile manners in software development projects. In those projects, the agile methods and the usage of its have not been structured. The ability and the level of usage have correlated to the knowledge of the project group, especially the project manager.

2.2 Agile Experiment and Implementation Method

The agile experiment and implementation method aims to improve the situation when there has not been a structured way of working in agile software development projects as mentioned in chapter 2.1. The idea of the method is to search the best ways to implement the desired outcome by continuous feedback analyzation and adjusting the target according to that. The method is suitable for all kind of a software development projects but especially it fits to the projects that may not have a clear vision or goal at the beginning. In addition, the method supports the rapidly made inexpensive experiments before committing to larger costs. [7]

The agile experiment and implementation method is part of the case organization's collection of development methods called KEHMET. The method bases to the British government model of the digital service design principles (GDS). [7] The other methodologies used when developed the agile experiment and implementation method are PRINCE 2, Safe, Scrum and Scrum of Scrums. [4] In addition, the Lean ideology is taken into account in the development of the method. [3]

The agile experiment and implementation method divides into two parts, the experiment and the implementation. Parts consists four phases called experiment, alfa, beta and live. The agile development process is the same in every phase of the method. [7] The phases, especially in the experiment part can be done individually without a need to implement the whole method. If the all phases of the method are used, then the progress is chronological.

In a software development project that follows the agile experiment and implementation method, the product owner's task is to evaluate the progress of the project. After each phase the product owner evaluates whether to continue to the next phase of the method, repeat the current phase or change to another development model, for example traditional waterfall model. A product owner presents the proposed option of the progress to the executive board that makes the decision to continue the project.

2.3 Agile Software Development Environment in Case Organization

Concerning the agile software development in the case organization, the agile experiment and implementation method is just a part of the agile way of working. The environment in which the method is used must also be taken into account.

Roles and Responsibilities

The current case organization's structure is built to support the traditional model of software development. The product owner role, that is the key role in the agile development, is not necessarily supported in sufficient level. Understanding the role and responsibilities of the product owner requires harmonization of the terms and concepts. The employees in other roles of software development should also be more familiar with the agile development method. In addition to the product owner, these roles are for example executive board, customers, users and technical support. If these roles are not aware of the methods used in the agile development, it might slow down the work of product owner. The product owner work will hamper if the executive board is not sufficiently familiar with the agile development to be able to steer the project. [8]

Framework Agreements

The case organization offers framework agreements to help the development projects to acquire expert services that is not available in-house. A framework agreement is ready-made environment to order services from predetermined vendors without a need of open market competitive tendering. In a framework agreement, there is set of conditions to an acquisition such as the offered services, prices and the terms of payment, immaterial rights and so on.

For the agile software development, the case organization offers two beneficial framework agreements. The first is the service design framework agreement. That offers an expert services and consultation in a service design and a user experience, which may be useful especially in the experiment phase of the agile software development project. The other one is the framework agreement of the open source software development. Via that is possible to acquire the software development and implementation services such as a user interface design and front-end and back-end coding.

Boundaries and Conditions

The agile experiment and implementation method is still under development in CITO. The method usage is limited for chosen software development projects and the CITO supervises the use of it. At the time of writing this Thesis, the method supported only the software development.

The framework agreements are not part of the agile experiment and implementation method. However, a recommendation is to use the framework agreements in projects which needs to acquire the expert services. Usage of the open source software development framework agreement is allowed only under the AOK supervision. AOK validates the orders based on the framework agreement before ordering. [9]

3 Method and Material

This section introduces the research plan and the model of the research. It offers an understanding about the different steps of the research and presents the process of the research. In addition, this section describes how the data was collected and analyzed in the research.

3.1 Research Plan

The plan of this research was divided in five steps. The Figure 1 illustrates the steps of the research plan and the aim of each step. The plan begins by setting up the objective and scope. That brings the focus to the research. After that follows the current state analysis of the agile software development in the case organization. The current state analysis bases to the interviews of the specialist and discussions with the employees related to the agile development in the case organization. In addition, the existing knowledge about the agile experiment and implementation method and some of the previous experiences of using it is presented in current state analysis.

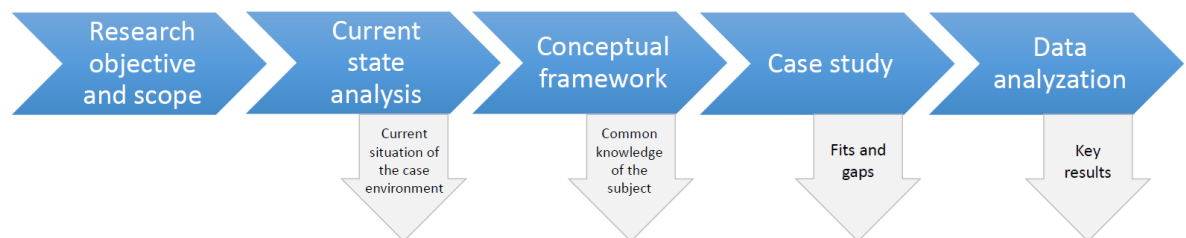


Figure 1. Research Plan.

After the current state analysis, the research proceeds to the conceptual framework. Purpose of the conceptual framework step is to offer a general knowledge about the agile software development and the techniques related to it. The conceptual framework is not taking into account all the agile development techniques, but the ones that are essential according to this research. Current state analysis and conceptual framework acts together as a starting point for the case study implementation.

Case study in the fourth step covers the implementation of the research. In the case study, the case project is carried out by using the agile experiment and implementation method. Structure of the case study is cyclical according to the research design presented in section 3.2. Case study implements the requirements and the tasks set to the product owner in the agile experiment and implementation method. The Fit-GAP analysis method presented in section 3.2. was used in the case study section to collect the findings of the research. After the case study, there is a data analyzation step. In that step the fit-gap analysis of the tasks in case study are collected together. Findings of the individual requirements are sorted out to larger groups according to the categories of the Fit-GAP analysis. The target is to analyse the gaps and gain the information needed to formulate the recommendations to improve the current situation.

3.2 Research Model and Process

The research model in this Thesis is case study. A case study as an approach is useful in order to gain more in depth information of the particular aspect. [10] A typical case study research method aims to answer the question “how” or “why”. The case study counts on the same techniques as many other research models. In addition, the case study includes direct observations of the subject events and people interviews that are not usually counted as a source of evidence in other models. [11]

The Fit-GAP analysis method is used in the case study section of this Thesis. The idea of the Fit-GAP analysis is identify where the existing or planned system meets the needs of the case organization. If the needs are met, then it fits. Otherwise, there is a gap. The Fit-GAP analysis not only present the answer does the current functionality fits or not, but it offers help to identify the causes of the gaps. In addition, with the Fit-GAP analysis it is possible to analyze the reasons of the gaps, and formulate and prioritize the problem solution. [12]

This research uses the cyclical process to go through the implementation of the case study. The cyclical way of progressing in the case study supports the phased structure of the agile experiment and implementation method. It offers needed steps for preparing, implementing and analyzing the tasks. In addition, cyclical process offers a point of decision making timely according to the requirements of the method presented in the section 2.2. Figure 2 presents the model of the cyclical research process.



Figure 2. Cyclical Research Process.

The cycle starts with the planning step where the tasks and requirements of the product owner in the present phase of the agile experiment and implementation method are offered as an input to the cycle. The tasks are listed to the Fit-GAP analysis template. In the second step, the tasks and requirements are implemented in the case project and the findings of the implementation are documented in the Fit-GAP analysis template.

The third step is to overview the findings of the Fit-GAP analysis. This step focuses to find out the gaps that may cause problems in the following cycles. Last step of the cycle is the evaluation. There one makes a decision according to the findings whether to start the new cycle within the next phase of the agile experiment and implementation method or re-do the current phase. In addition, if the results of the implementation shows up problems going further in the project one needs to evaluate the possibility to abort the project.

The cycles of the research process are fitted to the phases of the agile experiment and implementation method. One phase of the method is handled in one cycle. In addition, the project settings before starting the methods implementation is treated as a one phase and it makes one research cycle.

The Fit-GAP analysis is used for collecting and analysing the findings of the tasks implemented in the case study. The used template of the Fit-GAP analysis is following the structure presented in Appendix 1. Three of the columns consists the structure of multiple-choice. The Figure 3 illustrates the options of the multiple-choice fields of the template.

GAP Analysis	GAP Source	GAP Requirement type
<ul style="list-style-type: none"> • GAP = Full gap • PG = Partial gap • NG = No gap 	<ul style="list-style-type: none"> • MR = Method requirement • ORG = Organizational requirement • Other 	<ul style="list-style-type: none"> • Method • Product owner • Organizational • Technical • Other

Figure 3. Multiple-choice Fields in Fit-GAP Analysis Template.

The GAP Analysis column presents the status of the task after implementation. The GAP Source and GAP Requirement type columns are filled only if the finding was a full or a partial gap. The GAP Source and GAP Requirement type columns specifies the source and the nature of it. The comment column stands for the additional information or clarification to the task findings.

3.3 Data Collection and Analysis

The data in this research was collected from the multiple sources. The literature review utilized both existing literature and Internet sources. For getting familiar with the current state of the agile software development in the case organization, interviews and discussions were conducted with the employees related to the agile development in the case organization. In addition, one used the case organization's intranet websites, existing documentation related to the agile experiment and implementation method and earlier experiences of agile development in the case organization. After collection, data was analyzed by picking-up the boundaries and the requirements that sets the starting conditions for the case study research.

In the case study section, the data was collected from the organization's internal documentation, user survey, discussions with the people related to the case project, workshop with the user group and the implementation of the product owner's tasks. The data from the survey, discussions and workshop was collected and analyzed for using it as a material for the case study. The data gained from the case study implementation was collected to the Fit-GAP analyzation template and was analyzed by the Fit-GAP analyzation methods.

4 Conceptual framework

This section introduces the conceptual framework of the research subject. It aims to offer an understanding of the agile software development and the product owner's role in general level. The conceptual framework is not taking into account all the agile development techniques and methods, but the ones that are essential to this research. This means those methods and ideologies that the agile experiment and implementation method is based.

4.1 Overview of Agile Software Development

The agile development stands for the idea of being able to respond to the needs in fast changing and unpredictable environment. Concept of agile software development is a collection of methods and practices based to the values and principles presented in the Agile Manifesto. [13] The four values of Agile Manifesto are based to the idea of finding preferable ways of developing by implementing the software development and helping others in it. [14]

The nature of agile software development is delivering fast and flexible the value to the customer by acting iteratively. The working manners of agile software development are based to the delivery of small increments instead of large software launches and consists the continuous evaluation of the requirements, plans and results for being able to response the changes quickly. [15]

4.2 Agile Development Methods

There is multiple general well-known agile development methods for different agile software development purposes. These are for example Dynamic Systems Development Method (DSDM), Extreme Programming (XP), Feature-driven development (FDD), Crystal and Agile modeling. [16] However, this section introduces from the methods only the Scrum and Scrum of Scrums, which are used as a base of the agile experiment and implementation method.

4.2.1 Scrum

In the Scrum method, the focus is to build the product iteratively by using the fixed-length development periods called sprints. Good evaluation and the rapidly gained feedback from the product tests are required while used the short iterations. Visual artifacts, such as task boards and burndown charts used during the sprints acts as a motivator of the Scrum development team. [17] The approach of Scrum method is to concentrate highly collaborative working towards frequent delivery and continuous improvement. In addition, the responsibilities of the roles in Scrum are clearly defined. The artefacts used in Scrum are product increment, product backlog and sprint backlog. Additionally to these, the quality of the product is taken in to account by using the tool called definition of done. [18]

The structure of Scrum method consists three roles. The roles are Scrum Master, Product owner and Scrum Team. The Product owner's task is to bring the requirements of the product to the product backlog. During the sprint, the Scrum team implements the increment of functional product according to the requirements in product backlog. The role of Scrum master is to support the other roles in their work and eliminate issues that might be faced. [18]

In Scrum method, the structure repeated in every sprint includes four events that are sprint planning, daily Scrum, sprint demo and sprint retrospective. In the sprint planning there is defined what should be fulfilled in the upcoming sprint. During the sprint is held the daily Scrum that is 15 minutes meeting for synchronizing the Scrum team's work. The sprint demo is where the Scrum team is presenting the deliverables produced in the sprint. The sprint retrospective is for reviewing the successes and difficulties faced during the sprint for improving the working in the next sprints. [17]

4.2.2 Scrum of Scrums

The Scrum of Scrums integrates the work of several Scrum teams that are working with the same project. The aim of the integrations in Scrum of Scrums is to allow communication between the Scrum teams. By that is ensured that the team's software output integrates with the other teams outputs. That is needs especially when there is overlapping or needed a particular sequence of events. [19]

The Scrum of Scrums' meeting consist one person of each Scrum team. The Scrum team choose the person to be as ambassador in the Scrum of Scrums meeting. The ambassadors' role in the Scrum team may be the teams' Scrum masters or the technical contributor depending on the context. [20] The Scrum of Scrums meetings are held for coordinating the work of various Scrum teams. The ambassadors should be able to present the accomplishments after the last meeting, the possible occurred problems and their effects to the team's work. Additionally, there is presented the aimed accomplishments before next meeting. If there is found any interference problems of other teams that may effect to the team's work that is presented too. For keeping track of these, there is held a Scrum of Scrums' product backlog that is maintaining the chief Scrum master. [19]

4.3 Product Owner in Agile Software Development

The product owner in agile software development is responsible for that the development team's work produces as valuable as possible product. The product owner's role and responsibilities are managed by one person. To be able to succeed in the product owner work, the organization needs to have respect to the product owner's decisions. The product owner is only person managing the product backlog. If there is desired to present the changes in to a backlog prioritization that needs to be agreed with the product owner. [21] The product owner is responsible for offering the clarification of the product backlog items to the development team and deciding the items to be developed. [22]

Performing the product owner role, there are several main activities. Identifying and describing clearly the items in the backlog to offer an understanding of the problem and solution to the development team. By prioritizing the backlog items and decision-making according to that is done for being able to deliver the maximum result with less effort. Product owner evaluates is the item of the product backlog delivered sufficiently. [22] The product owner may implement the activities by itself or delegate those to the development team. Nevertheless, accountability of performing the activities remains with the product owner. [21]

4.4 GDS

The agile experiment and implementation method takes advantage of the Government Digital Service (GDS) of British government. GDS is a part of the British government Cabinet Office and its purpose is to work for the government digital transformation. That is done in area of digital, technology and data by supporting the departments in their transformation by building the platforms, standards and services with them. [23]

Work of DGS is open, agile and guided by a set of design principles. Starting point for GDS's work is always the users' needs. Additionally to the work done with the departments, the GDS work towards simpler and better public services in whole government. That relates the platforms building such as GOV.UK, ensuring the good quality and usability of the government data and supporting the departments' decision-making when acquiring the technology. [23]

4.5 PRINCE2

Projects in controlled environments (PRINCE2) is widely known method for project management. Using the best-practice approach of the PRINCE2 has been found to increase the probability of a project successes. However, using the PRINCE2 is not guarantee of a successful project. Instead, the PRINCE2 offers an aspects and ideas how to manage the project. One may use that while evaluating if there is deficiencies in the project management. [24] Structure of the PRINCE2 is based to the seven principles, processes and themes. Flexibility of the PRINCE2 enables the method's scaling for different type and size projects. [25]

The PRINCE2 principles presents the requirements and good practises to follow. When managing the project by using the PRINCE2, these seven principles needs to be used, unless it is not counted as a PRINCE2 project. Processes of PRINCE2 defines the needed steps for the project lifecycle. Each of the seven processes consists checklist where is collected the recommended activities and related responsibilities. Additionally it includes the guidance to tailoring the processes to a specific environment. [25] Themes of PRINCE2 describes the recommended ways of carry out the certain aspects of the project. Any of the themes could be used throughout all the processes and several themes may be used in one process. [24]

4.6 Lean

The main focus of the Lean is enterprise value stream. In the Lean, the consumer world is the content used as an input of the value stream when starting the development. By that the each activities done subsequent would add value to the customer. Lean avoids of waste in production that reduces the value and favour the continuous improvement to increase the value. Utilizing the Lean, the use of agile principles and ideas have better support in the project. The Lean is more about people and interacting than processes and tools. [26]

Applying the Lean principles to software development is gained a lightweight up-front architecture. That means there is reduced waste such as reworking, unnecessary artifacts and wait states. [26] First of the five principles of Lean is the value. That is to say the value of the product from the customer's point of view. Second is value stream. What that is defined the needed steps in the value stream by same time reducing the steps that are not producing value. After that is the principle of flow, that is for organizing the steps so the product is flowing smoothly to the customer. Fourth principle is pull, which enables the customer to pull the value from the stream activities. Last one of the principles is perfection that wraps up together the above principles. The principle of perfection is to repeat the process until have reached the state of value perfection where is no waste created. [27]

4.7 SAFe

The scaled agile framework called SAFe is an interactive software framework. With SAFe, the large enterprises may apply the practices of Lean-Agile and Scrum. Nature of SAFe is that it guides exactly what the organizations needs to be implementing. [28] SAFe framework is knowledge base that is available free online. With SAFe high amount of knowledge is described the roles, responsibilities, artifacts and activities needed in Lean-Agile development implementation. SAFe enables co-operation, collaboration, and delivery between several agile teams. Use of SAFe can be scaled from the small solutions to the complex systems environments. With scalability and configurability, SAFe can be adapted to different kind of an organization's needs. [29]

With the four out-the-box configurations, the SAFe enables support for wide range of different kind of development environments. The Essential SAFe configuration is the most simple configuration of SAFe and seen as a heart of the framework. Portfolio SAFe configuration is for aligning the portfolio execution to the organization's strategy. Large solution SAFe is used when developed large and complex solutions whit multiple agile release trains and service vendors. However, in the large solution SAFe is not needed a portfolio-level considerations. Full SAFe is for organizations that are building and maintaining the large integrated solutions and it contains all of the presented SAFE configurations. [29]

4.8 Summary

There are a number of different approaches and perspectives to implement the agile software development. It is important to choose the right methods for different kind of a software development needs and applying those according to the environment requirements. The methods, frameworks and ideologies presented in this section affect in the structure and content of the agile experiment and implementation method. By that, these are involved in the case study of this Thesis.

5 Implementing Product Owner Role – CASE Huvaja

This section describes the implementation of the case study. It starts with an overview to the baseline of the case study and presents the initial data of the project. Then the implementation of the case study is carried out and the findings are collected by following the research process presented in section 3.2.

The resource booking system, called Huvaja, a renewal project is the case study of this Thesis research. The service owner of the Huvaja resource booking system is located in the Administrative Division of the case organization. The project owner is located in the CITO of the case organization. The product owner of the project is located in the Kanslia ICT of the case organization. Other roles, resources and stakeholders of the project are introduced in the case study in the point they appear.

The tasks and requirements of the phases come from the agile experiment and implementation method. The product owner is responsible of the tasks. Exception to that is the phase that includes the tasks of setting up the project environment. That phase includes tasks to the several different roles for the case organization. Within the phases, the implementation of the tasks may proceed partially overlapping or the tasks may change the order of appearance. The case organization hoped for the product owner to pay a particular attention to the user experience and feedback from the users as part of the software development especially at the experiment part.

If the case study project needs to acquire the expert services, it follows the recommendation of the case organization to use primarily the service design and the open source software development framework agreements.

5.1 Project Setting

Target of the project setting phase is to build the project organization, liable employees to the key roles, define the key stakeholders of the project and set the boundaries such as timetable and budget. Tasks of the project setting phase in the case project are:

- Setting the product owner to the project
- Setting the steering group to the project

- Defining the in-house technical support needed in the project
- Resourcing the product owner
- Defining the key stakeholders of the project
- Product owner's familiarization to the project
- Project timetable
- Project budget
- Preliminary plan of the project for the steering group to get permission to start the project

In the project setting phase, not all tasks are directly the product owner's responsibilities, but the case organization is taking care part of them. The tasks in this phase should be fulfilled as precise as possible. However, some of the tasks may need specifying during the project.

Implementation of the tasks started with setting the structure and key roles to the case project and defining the key stakeholders. The structure of the project management in KEHMET is divided in four options [30]. In the case project the structure option three was used that consists the steering group and the project group. Key roles defined for the case project are the product owner and the members of the steering group. The members of the project group may vary during the development. However, the product owner is a permanent part of the project group. As KEHMET states [30], the other defined roles are an ICT architect, a solution architect and an information security and data protection responsible. If expertise of these roles are needed, these are available from the CITO for this case project.

Internal stakeholders defined for the case project at this point are a specialist group and a users group. The specialist group consists the specialist of the renewable information system. The specialist group is an important group especially in the discovery phase to produce input for the development. Specialist group acts also a pre-testing group for the Huvaja before releasing it to the users group testing. Users group consists the employees of the City Executive Office. The users group is involved in the project when collecting the information about the needs and desires of the information system features and functionalities. In addition, the user group may take a part to the testing of the information system in the implementation phase of the project.

External stakeholders defined in this point are service vendors that may be used via framework agreements. The information office of the City of Helsinki Executive office is taking care of the information of the development to the users. The project group handles the information inside the project organization and to the specialist group.

The product owner is a part of the steering group and the project group and operates as a link between the other actors in the project as illustrated in the Figure 4. Product owner acts as a presenter in the steering group meetings and keep the steering group up to date about the project progress. Product owner is authorized by the steering group to make decisions of the project operative actions in the project group.

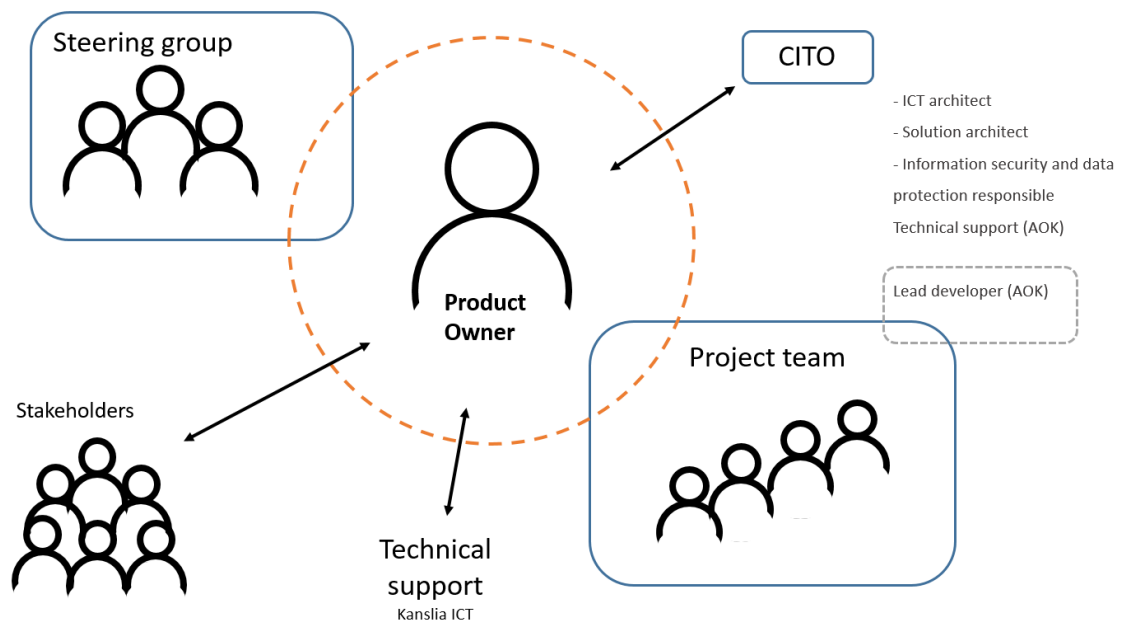


Figure 4. Links of Product Owner in Case Project.

To be able to carry out the case project, the product owner is familiarized to the role and responsibilities of the product owner, project environment and the used methods by the case organization. In addition, the product owner is instructed how to operate with the CITO and what kind of a project group and steering group roles there is in agile software development.

The product owner typically needs a technical support in the project. For the technical support, there is defined two actors: the technical support of AOK and the technical support of Kanslia ICT. The technical support of AOK is focusing to the software design and development and supporting the use of open source software development framework agreement. If needed, the AOK offers a leading developer to the project that guides the work of software developers during the project.

The technical support needed from the Kanslia ICT in the project relates to the technical infrastructure of the case organization's environment and the existing services that may need to be integrated to the Huvaja. In addition, the technical solutions such as authentication may need a technical support from the Kanslia ICT.

In the project setting phase, the steering group sets the timetable for the case study experiment part. It starts with the project setting followed by the discovery and alfa phases. The targeted time to complete the phases is four months. The timetable for the implementation will be evaluated after the experiment part is finished. In addition to the timetable, the budget of the project is set for the experiment part only. The budget of the implementation is evaluated as well after the experiment part.

Evaluation of the fits and gaps and findings of the tasks are presented in Appendix 2. The challenges in the implementation of the tasks in this phase was focused to the organizational aspects. The product owner's resourcing was complicated due the ownership of the information system. The agile experiment and implementation method is focused to the software development and therefore the product owner needs to be IT-oriented. Even if it is recommendable that the product owner is the same during the development and in the maintenance, there was made a decision to asset the product owner from the Kanslia ICT to the development project. If the information system development proceeds to the live phase, the product owner's role will be handed to the service owner.

Product owner's familiarization to the project environment was defective. The information about the current state of the renewable information system was available, but there was lack in the guidance to the use of agile experiment and implementation method. In addition, the available documentation was not in a sufficient level to gain enough knowledge about the working in agile software development environment and the organization's support for the product owner was not adequate. To understand the product owner's role in the agile software development in case organization's environment and the use of agile experiment and implementation method, the product owner would need a training from them, but that was not available. The solution to cover this problem was to use of the knowledge of the experts in the case organization that could help the product owner and share their expertise whenever needed.

The target of this phase was reached in sufficient level. The project group was build, responsibilities were pointed to the key roles and the boundaries were set. It was taken into account that gaps founded in this phase may bring challenges in the following phases, but they were not preventing to continue with the project. Steering group accepted the preliminary project plan and gave the permission to proceed to the project implementation.

5.2 Experiment

The nature of the experiment part is to observe the subject, identify needs and try out possible solutions. The experiment part is divided into two phases, discovery and alfa. Discovery phase covers the background investigation and collecting the requirements for the development. The alfa phase is for the prototype implementation based to the gained information of the discovery phase. After the alfa phase, the success of the experiment part is evaluated. According to the evaluation, the experiment part will be re-done, the case project proceeds to the implementation part or the case project is terminated.

Involving the users to the development is essential in the experiment part. In this case project that was done by using the user query and the workshop with the users to gain the requirements for the Huvaja alfa implementation and by user testing in the alfa phase to gain the feedback of the Huvaja prototype.

5.2.1 Discovery

There is described two main purposes of the discovery phase in the agile experiment and development method. The purposes are to create an understanding of the needs of the service and to define the options for the solution. The target of the discovery phase is to found out if the idea of the service is promising enough for the prototype implementation. The discovery phase is continued until the sufficient starting point to the alfa phase prototype development have been achieved. [7]

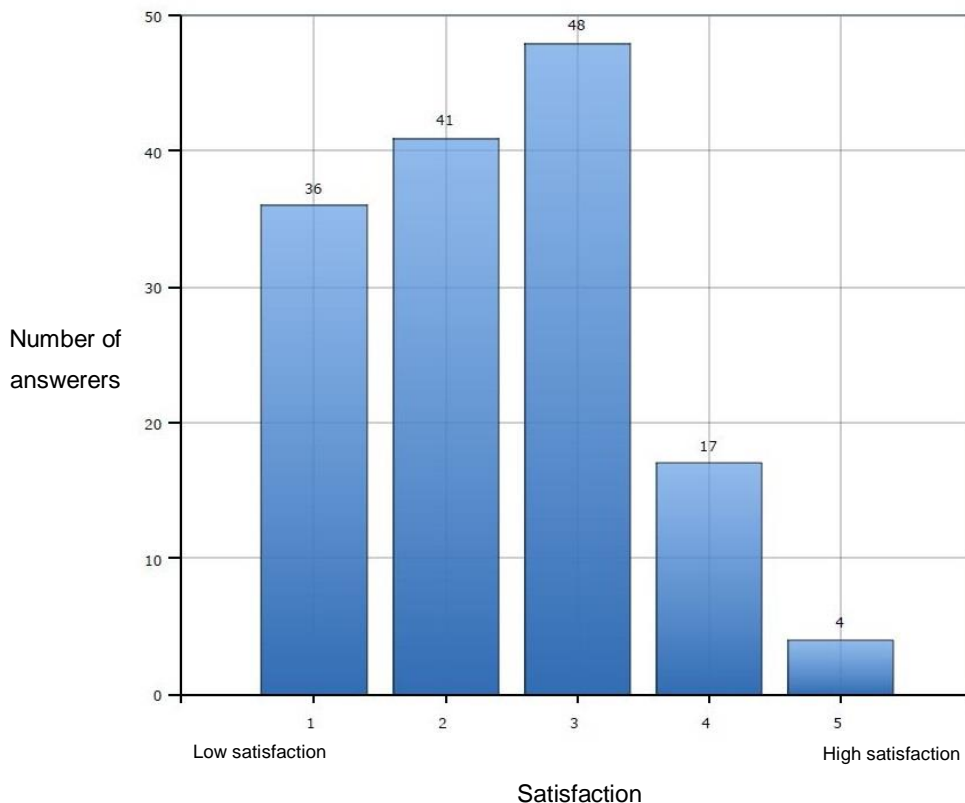
The discovery phase of the case project includes defining the purpose of the Huvaja, collecting the user needs and selecting the solution for alfa phase prototype implementation. After that is produced the vision of the Huvaja and the preliminary roadmap for the project. The discovery phase includes tasks to the product owner in the case project as follows:

- Defining and acquiring the organization's internal resources needed for the phase
- Current state analysis
- User query
- Analyzation of the user query
- Committing the specialist group and the key role users to the requirement specification
- Service design acquisition
- Formulating the roles and requirements of the Huvaja– workshop
- Selecting the user stories for the alfa prototype implementation
- Options of the solution to the alfa phase
- Vision canvas
- Preliminary roadmap
- Benefits comparison
- Phase ending presentation to the steering group

At the beginning of the implementation of the discovery phase, the needed organization's experts were collected in the project group. In addition to the product owner, these were the technical support, expert of the renewable information system and the service design expert. For getting to understanding about the current state of the renewable information system there was an analyzation about the available information of it such as description of the current system maintenance and the documentation of earlier pilot project including requirements defined at that point.

For gaining a wider view of the users' expectations and needs for the Huvaja, there was carried out the user survey to the users group. The user survey offered answers to the project about how satisfied the users are with the current system. The results of user satisfaction with the current system are illustrated in the Table 1, where one point stands for a low satisfaction and five points stands for a high satisfaction. The overall satisfaction with the current system was 2,4 out of 5. In addition, the user survey presented answers about how reliable users see the system and what are the key elements that should be developed.

Table 1. User satisfaction of current system.



After the user survey analyzation the project had preliminary guidelines for the requirements, but those needed to be specified in more detailed. For that purpose, one needed the users of the specialist group to work with the project group. For supporting the project group's and the specialist group's work there was acquired the service designing expertise work by using the service design framework agreement. The service design methods and consultant expertise was exploit by carrying out the user interviews for the users in the specialist group. The results of the interviews was analyzed and according to those, there were created the preliminary roles and requirements for the Huvaja. Then the preliminary roles and requirements was processed with the specialist group and the service design experts in the workshop. As an outcome of the workshop was defined the roles of the Huvaja and the requirements were formulated to the user stories. The product owner was evaluating the workload of user stories with the project group. After that, the product owner chose the user stories for the alfa phase prototype implementation.

For choosing the technical solution to the alfa prototype implementation there was a study about the existing solutions in the case organization's environment and discussions with the technical support of AOK. Based on these, it was decided to utilize the existing interface solution and create the new user interface for Huvaja. It was noted, that the existing interface solution might need a new instance for the Huvaja.

The evaluation of Huvaja benefits was done by comparing the current state to the aimed information system implementation. In addition to the cost-benefit analyzation, the user experience was an important beneficial aspect. After that, the vision was done for collecting the information gained in this phase to a one condensed form. The vision canvas illustrated in the Appendix 3 crystallized the needs and the potential users of the information system, the proposed technical solution and how the information system produces value to the users. Last in this phase was created a road map for the information system implementation.

The fits and gaps of different tasks in discovery phase are presented in Appendix 2. As a finding of this phase, the roles and requirements of the Huvaja were produced successfully. However, defining and acquiring the organization's internal resources in the beginning of the discovery phase was difficult because there was not a clear understanding what expertise was needed during the phase. Therefore, the need of internal resources was discovered only during the phase.

Committing the specialist group to the roles and requirements specification was challenging because of the product owner's insufficient knowledge about the specialist role users. In addition, the expert role user's understanding of working with the agile manners was not in a sufficient level for fluent working. The product owner was helping and directing the specialist group closely.

Acquiring the expertise work via service design framework agreement succeeded, but the guidance to use the framework agreement was insufficient. Product owner needed support from the CITO experts to define the appropriate way of using the framework agreement. In addition, producing the documentation in the discovery phase was partially challenging. There was not enough information on how to use the vision canvas presented in the agile experiment and implementation method. The case organization could not support the use of vision canvas so the project needed to acquire the support from the service design experts.

The service design and focus in user experience was one of the key elements in the discovery phase and using services from service design expert contributed the project. By collecting and evaluating the users' needs and formulating those with the expert group, gained the user stories for the alfa prototype implementation. The vision canvas of the project benefit the comparison and the preliminary idea of proceeding to the alfa phase prototype implementation was presented to the steering group. The steering group decided that the discovery phase had produced the needed information to start the implementation of the information system and gave the permission to proceed to the alfa phase.

5.2.2 Alfa

The target of the alfa phase is to try out the solution of the service by implementing the prototype of it. The aim of the solution prototype implementation is to gain feedback from the users that is the prototype producing the desired outcome and leading the development in the right direction. In addition, the alfa phase should reveal the possible challenges in the solution implementation suitability to the case organization's technical environment and in the planned service processes. The alfa phase can be started when the steering group have approved that the discovery phase have properly ended. To succeeding in the alfa phase, the main metric is the user satisfaction to the prototype. The user satisfaction rate should be at least 60 percent. [7]

In the case project, the alfa phase covers the implementation of the Huvaja prototype and users testing of it. The user stories selected in the discovery phase defines the functionalities implemented in the prototype. The specialist group acts as a test group in the alfa phase prototype testing. The alfa phase tasks to the product owner in the case project are:

- Implementation plan
- Defining and acquiring the organization's internal resources needed for the phase
- Using the Kanban
- Utilizing existing data and systems
- Status presentation to the steering group
- Open source software coding acquisition
- Setting up the test group
- Service design acquisition for creating the test model
- Setting up the project group working and reviews
- Organizing the testing of the prototype
- Analyzation of the test feedback
- Analyzation of the technical solution of the prototype
- Analyzation of the information security of the prototype
- Phase ending presentation to the steering group

The alfa phase started with panning the phase implementation and defining the organization's internal resources to the phase. In the implementation plan there was identified the need of acquiring the open source software coding work for the prototype implementation as well as service design expert to work for creating the test model. The project group was selected to be same as in the discovery phase but additionally included the internal resource from AOK for user interface designing. In addition, the project group was completed with one acquired developer.

The Kanban board was used to arrange the user stories that were selected in the discovery phase, to the tasks for implementing the Huvaja prototype. The tool used for managing the tasks was Trello application. After discussions with the technical support of AOK, there was made a decision not to build the integrations in the alfa phase. The need of integrations was identified, but the functionalities of the prototype could be produced and tested as a standalone implementation. After these, the implementation plan and the choices made for the alfa phase was presented to the steering group. The steering group approved the plan and acquisitions for the alfa phase.

The open source software coding acquisition was made using the framework agreement. After that was settled the practices for project group working and the reviews of the prototype during the implementation. The prototype testing was set to carry out after the implementation. For the agile software testing, there was not existing practice in the case organization. Therefore there was produced the testing model presented in Appendix 4. The testing model was produced so that it serves also testing in the implementation part of the case project. For creating the testing model and producing the testing plan, there was acquired the service designing via service design framework agreement.

The prototype implementation took four weeks. After two weeks, there was kept the review with the project group. During the prototype implementation, the test group was set and guided to the testing. The test group in the alfa phase prototype testing was the specialist group. After four weeks development, the implemented prototype was reviewed with the specialist group. After review, the test group tested the prototype as an individual testing by following the test model.

Feedback of the test group was captured and then analyzed. The technical solution validity and the security of the prototype was analyzed according to the information gained from the AOK technical support about the code validity and the security aspects of the implementation.

Fits and gaps of the alfa phase tasks and findings are presented in the Appendix 2. Organizing the internal resources and producing the phase implementation plan was carried out successfully. The gaps was founded using the Kanban method and Trello application. These were not familiar to the product owner and there was not guidance in the case organization on how to use them. For using the Kanban method and Trello application the product owner needed support from the CITO experts and AOK technical support. The existing data and system utilization was challenging due it was not clear in the beginning of the phase that what data or system integrations might be needed in the prototype implementation and is there expertise available to carry out the possible integration.

The open source software coding acquisition contained a gap because there was no guidance available for the product owner of using the framework agreement and what information the invitation for tenders should contain. The product owner gained the support for producing the invitation for tenders from the CITO experts. The service design acquisition for creating the test model was done successfully, but organizing the tests and the test group was partially challenging because the test period was short and the product owner had to fit the timetables of the users in the test group to the testing schedule

The phase ending presentation to the steering group included results of the user testing of the prototype. The key result was that 88 percent of the test group was satisfied of the prototype. In addition, presentation offered an evaluation of the implemented prototype technical validity and security and the used time and budget of the experiment part of the case project. The steering group considered that the experiment part of the project had produced the needed information for making decisions for next steps. The steering group decided that the case project proceeds to the implementation part.

5.3 Implementation

The target of the implementation part is to proceed the development started in the alfa phase, produce the usable software, and ensure the maintenance and lifecycle of it. The implementation part includes the beta phase for the software development and the live phase for releasing, maintaining and deactivating the software.

In this case project, the information system development in the beta phase bases to the prototype implemented in the alfa phase. If the project proceeds to the live phase, there is done the transition of the product owner's role from case project product owner to the service owner. This case project live phase does not include the deactivation of the information system. The case project ends at the latest when the maintenance of the Huvaja has been ensured and the responsibility of the information system has been transferred to the product owner.

5.3.1 Beta

The beta phase of the agile experiment and implementation method is for the actual implementation of the service. It starts with the investment decision and proceeds to the implementation of the service. The implementation is done by following the minimum viable product (MVP) ideology that is focusing the features that are the most beneficial for users. User stories selected for the beta implementation should pay attention to the information security features and non-functional requirements that take into account risk analysis and security of the service. The targeted outcome of the beta phase is to produce a service that is mature enough to be released to the production environment. [7]

In the case project, the beta phase specifies the vision of Huvaja and defines the scope of the implementation. According to that the budget and timetable for the implementation is set. The beta phase implements the software and expands the testing to a wider group of users. Product owner's tasks in the case project's beta phase are:

- Selection and prioritization of the use cases for the implementation
- Implementation plan
- Defining and acquiring the organization's internal resources needed for the phase
- Specifying the vision canvas
- Specifying the roadmap
- Selection of the architectural solutions
- Status presentation to the steering group
- Open source software coding acquisition
- Building a Scrum team
- Setting up the conditions for the definition of done
- Organizing the product backlog
- Setting up the sprint planning, reviews and retrospectives

- Observing the development and evaluating the output according to the definition of done
- Organizing the testing for limited group of users to test the Huvaja during the development
- Analyzation of the feedback
- Status presentation to the steering group
- Organizing the data validation for the production-like open beta testing
- Ensuring the support for the Huvaja during the open beta testing
- Ensuring the support for the integrations during the open beta testing
- Organizing the open beta testing
- Analyzation of the beta test feedback
- Organizing the review of the information security and code quality of the Huvaja
- Phase ending presentation to the steering group

In the beginning of the beta phase was selected and prioritized the use cases for the Huvaja implementation. The use cases selection based to the user test feedback of the alfa phase prototype testing. The prioritization of the use cases was done by following the MVP idea by evaluating which features may produce most benefit for the users.

After use cases selection there was done the implementation plan for the beta phase. In addition, the vision canvas and roadmap was specified based on the information gained from the experiment part. Next was defined and acquired the organization's internal resources for the phase. In addition to the product owner, the project group included the technical support from the AOK. External resources will be added to the project group if needed. At this point, there was not yet paid attention for the possible need of internal resources for the open beta testing.

In collaboration with the AOK technical support there was selected the technical architecture and the solution for the Huvaja implementation. In addition, the needed integrations was defined. Technical support of Kanslia ICT was needed when the integrations are implemented. After that the implementation plan, vision of the Huvaja and the roadmap was presented to the steering group. Steering group accepted the plan of the Huvaja implementation and set the budget and the timetable to the implementation. At this point, the timetable was only for the implementation and the timetable for the open beta testing will be defined after the Huvaja implementation is done.

Next step was the open source software coding acquisition for the implementation. The acquired open source software coding experts was included to the project group. The suggested method for beta phase software implementation was according to the agile experiment and implementation method Scrum. The project group was build according to that guidance. The Scrum team in this case project included the product owner and the Scrum team. The Scrum team consists a leader developer, another developer and a user interface designer from the AOK and two acquired developers. The Scrum master was not set and the tasks of Scrum master role was divided between the product owner and the leader developer.

The Definition of Done (DoD) was set for the features that will be implemented in the beta phase. DoD included the demand of user satisfaction in user tests and the technical requirements such as code testing and architectural demands. After that there was organized the development cycles called sprints. Sprints length was decided to be two weeks including a sprint planning, a review and a retrospective. After every sprint there should be DoD approved features ready for the user testing. The product backlog for implementation was built in collaboration with the Scrum team.

In to every sprint, the Scrum team chose the features from the project backlog to be implemented. The sprint backlogs and the project backlog were held in a Trello application as the Figure 5 illustrates. During the Huvaja implementation, the product owner was observing the development and evaluating the output according to the DoD. The product owner was guiding the Scrum team and making a backlog prioritization when needed.

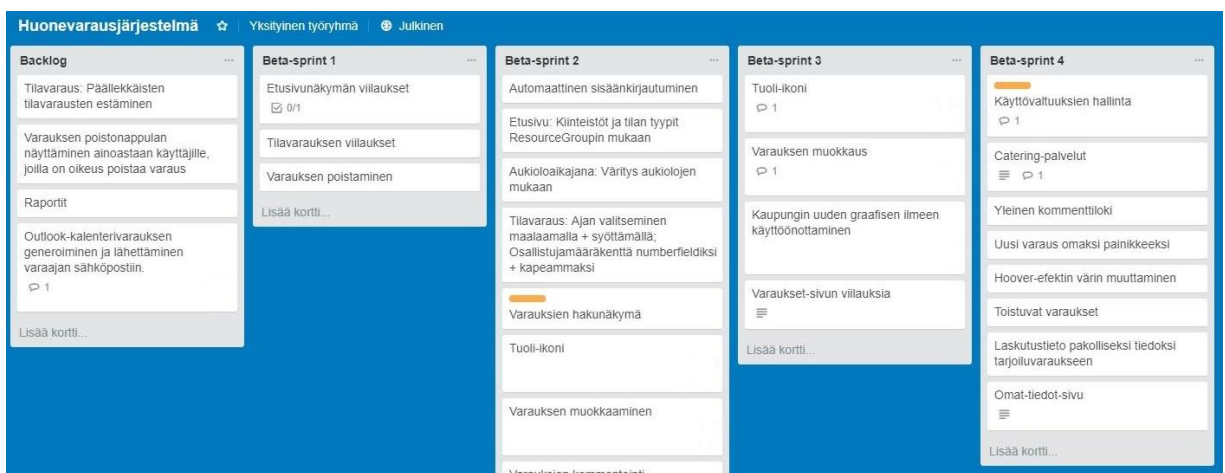


Figure 5. Project Backlog and Sprint Backlogs in Trello Application.

In the beta implementation, there was an organized limited group of users to test the features during the development. The test group consists mainly the users from the specialist group, but in addition to that a few users from the user group. After each sprint, the product owner informed the test group about the new features of Huvaja. The testing was continuous and users could provide feedback at any stage. Product owner collected and analyzed the feedback. After the implementation of Huvaja was done and tested, the status of the project was presented to the steering group. Presentation included additionally the implementation outputs and the test users' feedback. Steering group was satisfied to the Huvaja implementation and allowed the project continue to the open beta testing. The open beta testing environment was agreed to be production-like, but with a limited data. The duration of the open beta testing was set to two months.

Building the open beta testing was started with organizing the data validation. The test data used in the alfa and beta implementations and testing needed to be cleaned. After that, the valid data was imported to the Huvaja for the production-like testing. Next was ensured that the Huvaja and the integrations of it had a sufficient technical support during the open beta testing. In addition, the contact channels of the technical support and processes of problem solving was set. To the Huvaja was build the feedback feature for collecting the open beta testing feedback. Then the open beta testing was informed to the test users. The open beta testing was available for the user group that consists all the City Executive Office users.

Feedback of the open beta testing of Huvaja was collected and analyzed during the testing. When problems were observed, the product owner forwarded the information to the needed technical support. Observed problems were solved and the users' development proposals were collected. At the same time there was organized a review of the information security and code quality of the Huvaja. The review was done in collaboration with the AOK technical support.

The fits, gaps, and findings of the beta phase are presented in the Appendix 2. Tasks for specifying the Huvaja before implementation was carried out successfully. The selection and prioritization of the use cases was challenging because there was no clear understanding of the wideness of the implementation at that point. The product owner proposed the selected use cases to the steering group. The steering group made the decision of the wideness of the Huvaja implementation and the product owner fitted the use cases to that. In addition, acquiring the internal resources for the phase was found challenging due to it was not clear at that point if there was need for more support for integration building.

Open source software coding was acquired successfully, but there were difficulties to stay within the budget of the back end coding of the interface. That was solved by doing collaboration with another service that uses the same interface as Huvaja. Building the Scrum team, organizing the backlog and setting up the conditions for DoD was challenging because of the product owner's inexperience. For Scrum team building, there was information available for the product owner from external sources. The backlog was organized in collaboration with the AOK technical support. For defining the DoD there was support available for the product owner in the CITO.

The technical support for Huvaja for the open beta testing was organized successfully. However, organizing the support for integrations and data validation was found challenging. The processes for validating and importing the data were not clear and the service vendors' contact persons for integrations support were not set. For solving these, expertise from Kanslia ICT was needed. The open beta testing and feedback analysis was carried out successfully. Organizing the review of the information security and code quality was problematic due to the lack of product owner's knowledge about the case organization's instructions for that. The support of CITO and AOK helped to carry out this task.

After the analysis of the open beta testing feedback and the technical review, the results were presented to the steering group. Huvaja did not include open problems to be solved, the integrations functionality and support were ensured and the feedback of the open beta testing was indicating a sufficient user satisfaction.

The budget of the beta phase was kept. Timetable was delayed, but it did not affect the overall project. According to that information, the steering group stated that the Huvaja is ready to be released for the production. The steering group's conditions for going live phase was to arrange an external technical maintenance for the Huvaja and the product owner responsibilities transition from the project to the service owner.

5.3.2 Live

The live phase of the agile experiment and implementation method is for finishing and releasing the service. In that phase the service is taken under the maintenance and support and its agreed service levels (SLA) are monitored. Under the live phase can be done only updates and minor development to the service. In case there is a need for a larger development of the service, one should initiate a separate development project. Live phase can be started when the steering group have approved the service to be released and the possible problems occurred in the user testing have been solved. Life phase for maintaining the service can took for years and it ends to the service deactivation. [7]

The live phase of the case project ensures the technical support and produces the needed documentation for Huvaja maintenance. In addition, there is acquired the external service vendor for maintaining the Huvaja according to the requirements from the steering group. In this phase is defined the SLAs for the service and ensured the support for the integrations. The product owner's role is handed over to the service owner at the end of the phase. In the live phase of the case project the tasks of the product owner are:

- Transition plan from beta to live
- Defining and acquiring the organization's internal resources needed for the phase
- Maintenance plan of the Huvaja
- Defining the service level
- Price inquiry for the maintaining the Huvaja
- Status presentation to the steering group
- System maintenance acquisition
- Technical review of the Huvaja
- Architectural review of the Huvaja
- Documentation of person register and data protection of the Huvaja

- Ensuring the collection of the continuous customer feedback
- Ensuring the integrations maintenance
- Ensuring the organization's internal processes for Huvaja maintaining
- Continuity and recovery plans
- Organizing the data validation
- Ensuring documentation of the information system structure, maintenance policies and agreed practices for the service owner
- Ensuring the Huvaja technical transition to the external service provider
- Transferring the Huvaja responsibilities from project product owner to the service owner
- Project ending presentation to the steering group

The live phase started with producing the plan for transitioning from beta to live. The organization's internal resources to the phase consists service owner's experts for defining the needs of maintenance and the technical support of Kanslia ICT and AOK for the technical information system transition from beta to live. In collaboration with the service owner's experts there were defined the targets of the maintenance and created the maintenance plan. In addition, the desired SLA was defined. Based on the maintenance plan and the desired SLA, the price inquiry for the Huvaja maintenance was produced and sent to three service providers.

After receiving the offers of the Huvaja maintenance from the service providers, the transition plan, maintenance plan and the offers of maintenance were presented to the steering group. The steering group accepted the transition plan and evaluated that the received offers of Huvaja maintenance was eligible. The service owner made a decision of the service provider to the Huvaja maintenance. The service owner is responsible for the cost of the maintenance. The steering group did not set the budget for the live phase because no expert work acquisitions were needed. Tasks of the live phase will be carried out by internal resources.

The Huvaja maintenance acquisition was prepared and then the ordering documents were send to the service owner. Service owner made an order of Huvaja maintenance to the selected service provider. In addition, the service owner pointed the product owner from the City Administrative Office to be responsible of the Huvaja in maintenance phase.

After that was started the preparations for the Huvaja release. First was carried out the technical and architectural review of the Huvaja. These were done in collaboration with the experts of the CITO and the technical support of AOK. The Huvaja was added to the City Administrative Office's information system portfolio by Kanslia ICT and the needed data protection validation was done. It was agreed, that the maintenance product owner is responsible for producing the continuity and recovery plans according to the continuity and recovery processes in the City Administrative Office.

When the Huvaja was technically validated and added to the information system portfolio, there was ensured the environment documentation and the internal processes. That included the structure and integrations of Huvaja as illustrated in the Figure 6, maintenance policies and agreed practices. In addition, that included how to ensure the case organization's internal processes for Huvaja maintenance such as the agreed information channels in problem situations. After that, the product owner ensured the feedback feature built in the beta phase was in use and the gained feedback was directed to the maintenance product owner.

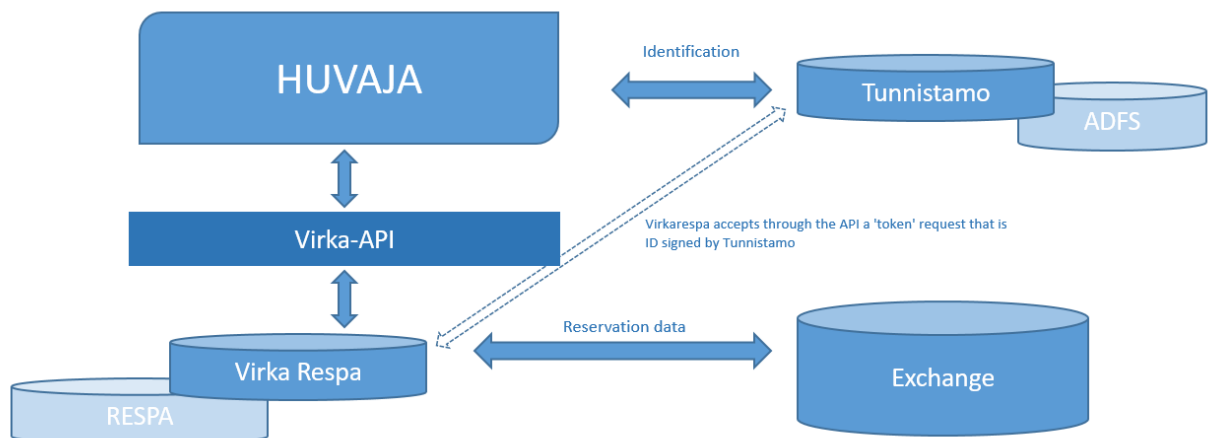


Figure 6. Structure of Huvaja and Its Integrations.

Before releasing the Huvaja, the data of it was validated. The data needed from the old resource booking system was integrated to the Huvaja in collaboration with the AOK technical support. Data specifications needed for the integrations was done with the technical support of Kanslia ICT. The product owner was ensuring the technical transition of the Huvaja. The transition was done between the technical support of AOK and external service provider. At the same time the responsibilities of Huvaja was transferred to the maintenance product owner. After that, the Huvaja was released.

As presented in the fits and gaps of the live phase in Appendix 2, the acquisition of the Huvaja maintenance was carried out successfully. However, preparing the acquisition faced challenges while defining the internal resources to the phase. The service owner could not set the product owner of the Huvaja maintenance at this point. The product owner of the Huvaja maintenance was set just before the technical transition of the Huvaja. Therefore defining the requirements for Huvaja maintenance was carried out by two users of the specialist group that acted as representatives of the service owner. For the price inquiry there was no framework agreement to use. In addition, there were not available examples or guidance of carrying out the price inquiry in the case organization. Support of the CITO was needed to define the content of the price inquiry.

Ensuring the technical reviews and needed documentation of the Huvaja was carried out mainly successfully. The integrations maintenance was found challenging due the roles and responsibilities of the problem solving process were not set. That was solved by creating the processes in collaboration with the integrated systems' service providers and the technical support of Kanslia ICT.

The experiment and implementation method in the case organization did not offer support or guidance to the transition of the product owner role. For that need, the case project produced in collaboration with the CITO a template called "checklist for the transition" to support the transition of the product owner role. The checklist for the transition is presented in the Appendix 5 and the needed documentation such as the maintenance policies and agreed practices for the maintenance were used in the transition of the product owner role. Ensuring the technical transition of Huvaja was found challenging due the lack of information about the responsibilities between the technical support of the AOK and the external service provider of Huvaja maintenance. That was solved by communicating with the technical support of AOK about the needed tasks of the technical transition, assigning the tasks and monitoring their implementation.

A project ending presentation to the steering group was held after transitions of the technical support of the Huvaja maintenance and the product owner role. The presentation included an overview of the case project, an evaluation of the case project outcome and gained benefits, the lessons learned, the current status of the Huvaja maintenance and proposals for the future development. The steering group stated that the case project had achieved its objectives and project was ended with a steering group decision.

The product owner was able to carry out successfully the case project by following the agile experiment and implementation method. However, the findings gained in the case project revealed that help was needed from the CITO and other case organization's experts in several tasks. Some of the tasks were challenging due the lack of information, common practices or knowledge. It is recommendable to analyze these findings and do improvements to increase the product owner's ability to work in agile software development projects.

6 Findings Analyzation and Key Results

This section analyses the fits and gaps and the findings gathered in the case study in section five. The aim of this analyzation was to identify the main development needs in agile software development environment in the case organization. The scope of the analyzation is to find out targets for improving the case organization's support for the product owner and the product owner's role in agile experiment and implementation method. At the end of this section is presented the key results of the analyzation.

Carrying out the case study presented in the section five, there was 78 tasks for the product owner. As illustrated in the Figure 7, the overall success of the tasks was 63 percent. However, it can be concluded that there is a need for some improvement in the role of the product owner in the case organization, as 37 percent of the tasks contained deficiency.

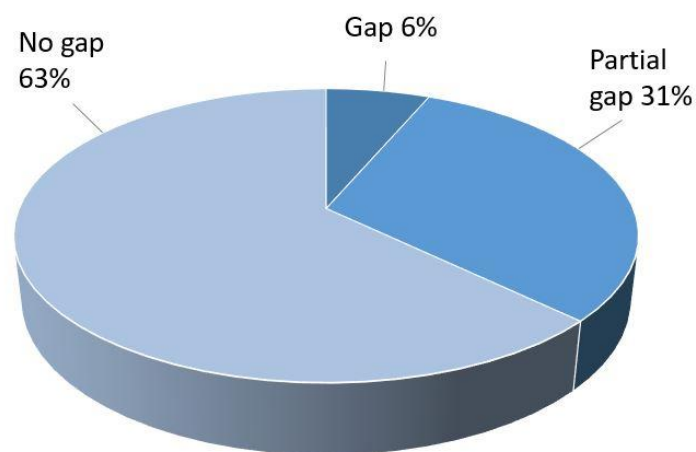


Figure 7. Fit-GAP Ratio of Case Study Tasks.

Analysing the gaps of the tasks, one was able to found out that the source of the full gaps was related four times to the organizational requirements and one time the method requirement. That points to the lacks of organization's ability to support the product owner's work. As the Figure 8 presents, the partial gaps were found equally from the method requirements and organizational requirements. Only in a minor part of the tasks the source was other than these. This reveals, that the improvements of the product owner role should be pointed to the case organization's support and the product owner's ability to use the agile experiment and implementation method.

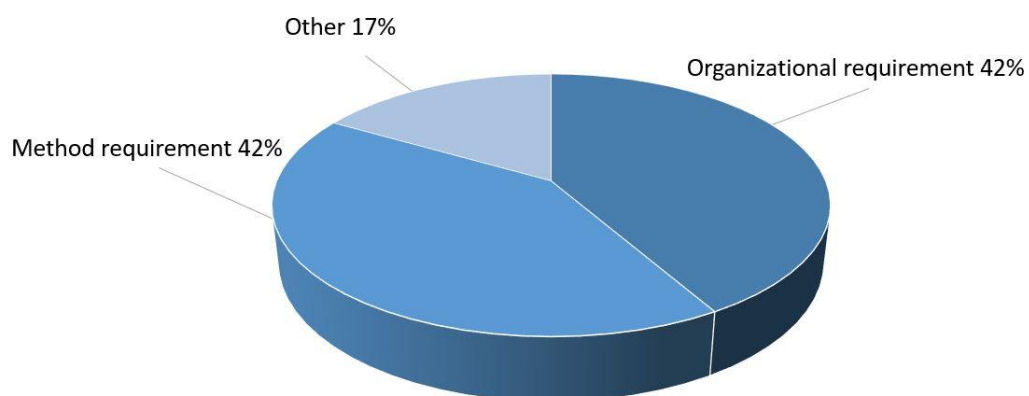


Figure 8. Sources of Partial Gaps.

The improvements were focused to the organizational and method requirements based on the results of the gaps analysis. After that, the findings of tasks related to the organizational or method requirements were reviewed. Findings were repeating certain themes. These themes were used in key results formulation.

As a key results of the organizational requirements were found that the product owner was not familiarized to the case organization's agile software development environment or the usage of the agile experiment and implementation method. Additionally training for the product owner's role or agile experiment and implementation method was not available and the product owner was lacking resources to carry out the tasks. The responsibilities of different agile roles in the case organization was not clarified and it was not clear what technical support the AOK and the technical support of Kanslia ICT was offering.

The key results of the method requirements were that there was challenges of using the required templates and the guidance of their use was missing or insufficient for supporting the product owner's independent working. There was not offered particular tools for supporting the working in agile software development project. The used tools were not easy to find and their use was not guided. A continuous support of the product owner was missing and there was no peer support available to help the product owner to use the agile experiment and implementation method. In addition, examples and lessons learned of earlier agile software development projects was not offered. In the key results was additionally noticed that the product owner's work was hampered by scattered documentation and lack of information.

7 Recommendations

This section offers recommendations based to the key results of the data analysis done in the previous section. Recommendations are focused to improve the product owner's ability to use the agile experiment and implementation method and carry out the agile software development projects in the case organization's environment.

7.1 Resourcing and Training

To be able to carry out projects in the case organization's agile software development environment, the product owner needs a sufficient support before and during the project. For helping the product owner to succeed with the project, it is important to make sure that the product owner has necessary knowledge and there are enough resources available. The recommended improvement areas for that are the familiarization, training and resourcing illustrated in the Figure 9.



Figure 9. Improvement Areas of Product Owner Support.

Familiarization

Product owner's familiarization to the case organization's agile software development manners and environment is one of the recommendations. Being familiar with the case organization's manners and environment helps the product owner to act in the agile software development projects. Thereby, it may improve the success of projects and raise the quality of the agile software development in the case organization.

For familiarization, one could produce a model that contains the most essential case organization's manners and practices for agile software development. In addition, the model could include guidance on how to use the key documentation of the agile experiment and implementation method such as the vision canvas. The product owners' familiarization could be implemented with discussions or by giving the documentation of the needed information to the product owner. Additionally there could be familiarization workshops in a case there are several product owners to be familiarized.

Training

One recommendation is to offer a training of agile framework especially to the less experienced product owners. By training one is able to gain the sufficient general level knowledge for acting in the agile development environment and the role of product owner in it. A certified product owner training could be a requirement for the product owners and it should be offered by the case organization. Additionally it would be beneficial to offer also training for other agile software development roles and for those who are working in any position in the agile software development in the case organization.

After the employees have gained the general level knowledge of the agile framework, all product owners should be trained to use the agile experiment and implementation method. By that, one can ensure the common understanding of the model usage. Additionally could be offered training about the essential practices and methodologies used in the agile experiment and implementation method such as Scrum and Lean.

Resourcing

When the case organization sets an employee to the role of product owner, one should take care of sufficient resourcing. The nature of the agile working commits the product owner more to the practical work of the project than the traditional project manager role. That should be taken in to account when assessing the required working time and effort for the product owner. In addition, the other key roles of agile development should be defined and resourced sufficiently to the projects for supporting the product owner's work. For gaining the needed expertise to the project, the case organization should offer a more flexible way to form the agile teams within the current division and between the divisions.

In the resourcing, it is recommendable to take in to consideration the development and operation culture (DevOps). If the case organization espouses the culture of continuous development, there should be created the continuous role of the product owner to the service. In addition, one could consider what agile development roles the case organization should have as internal resources in the future. For example having own service design expertise could gain benefit for the product owner's work in the case organization.

7.2 Tools and Support

As important as training and resources is to have sufficient tools and continuously developing networks for supporting the product owner's work. A product owner network, tools and methods and technical support illustrated in the Figure 9 are recommended areas to be developed for improving the product owner's support.

Product Owner Network

One of the recommendations is to build a product owner network. The idea of the product owner network was raised on the need of the product owners peer support in the case project. However, the peer support was not easy to find. The purpose of the product owner network is to collect the product owners from all divisions of the case organization together for sharing the experiences and lessons learned. Product owners with more experience could mentor and guide the less experienced product owners in their work. Regular meetings and a shared communication channel are recommended ways to maintain the network.

One target of the product owner network is to solve the practical issues related to the working as a product owner or using the agile experiment and development method. Alternatively the product owner network can escalate the issues to the CITO when needed. The product owner network can also develop the role of product owner together with the CITO to fit more proper for the case organization's agile design and development needs. Addition to that the maintenance and development of the product owner handbook can be also seen as one of the product owner network's task.

Tools and Methods

When acting as a product owner in projects, one needs a variety of tools and methods. To harmonize the case organization's agile software development environment and practices, there is recommended to exploit the unified tools and methods. Product owners would benefit of the selection of recommended tools and methods that are supporting the use of agile experiment and implementation method and proven to the case organization's environment. The tools and methods should be easily available for the product owners and include the needed guidance of their use.

When there is a need for a new tool or a method, that could be validated first by experts of the CITO, the technical support of the AOK or the product owner network. After that, the tool or method could be tested for example in the agile software development project or in software maintenance tasks. If the tool or method is found suitable, it could be added to the case organization's selection of recommended tools and methods.

Technical Support

The roles and responsibilities of internal technical support should be clarified. The technical support is essential for the product owner in agile software development. The case organization should be able to provide to the product owner information on what technical support is available and which one it produces. For that, responsibilities of technical support should be divided between the technical support of AOK and the divisions' technical support. Especially the technical support of AOK should be conceptualized for being able to offer equal service for all agile software development projects in the case organization. In addition, there should be a clear understanding when one needs to acquire external technical support.

The technical support's familiarization to the agile experiment and development method is recommendable. Getting familiar with the case organization's agile software development environment and the product owner's role in agile software development project improves the technical support ability to understand the product owner's needs of the technical support.

7.3 Handbook

One of the recommendations is a handbook for the product owner. Purpose of the handbook is to collect the useful information and guidance for the product owner into one place. The handbook is not a collection of the case organization' documentation, but it is offering a view to the information available. It guides the product owner to find the needed documentation, examples, practices, tools and support. The handbook offers information and guidance in three phases of the product owner's work as presented in the Figure 10. In addition to these, the handbook includes common information about the product owner role in the case organization.

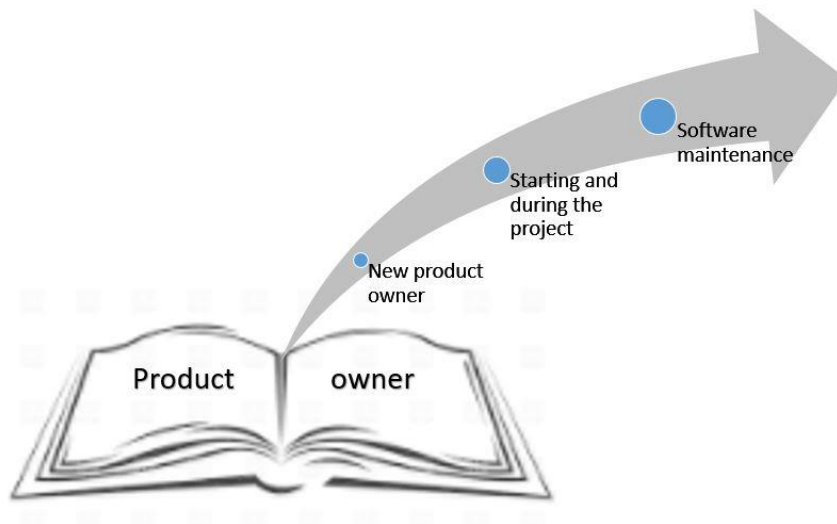


Figure 10. Structure of Handbook.

For a new product owner, the handbook offers the guidance to the agile software development environment in the case organization. It presents information about how and where the new product owner can get training to the role and where to find needed information about the support to familiarization and resourcing. In addition, to get the new product owner familiar with the agile software development environment, the handbook presents the examples of the agile software development projects in the case organization and lessons learned of them.

Supporting the product owner while starting the project and during it, the handbook offers a guidance to use of the agile experiment and implementation method and the framework agreements. The handbook proposes the agile design and development tools and techniques for the product owner that are validated to be suitable in an agile software development in the case organization's environment. In addition, there is offered information about the support for the product owner such as technical support available and the peer support via product owner network.

For the software maintenance phase, the handbook offers to the project owner guidance to the different options for maintaining the software and evaluating the software's lifecycle. In addition, the handbook presents the information about what needs to be taken into account in transition between the development and maintenance.

8 Discussion and Conclusions

This section summarizes the Thesis. It offers a case organization's feedback of the research and gained results and formulated recommendations. Additionally this section provides an evaluation of the validity and reliability of the research and conclusions of the Thesis.

8.1 Summary

The objective of this Master's Thesis was to research the product owner's role in case organization's agile software development environment. The aim of the research was to answer the question if the product owner is able to carry out the case project by using the agile experiment and implementation method. With this case study, the case organization could gain information about how the agile experiment and implementation method suites for the product owner's working environment and suggestions how to improve the support for the product owner in agile software development in case the organization.

For achieving the research object, the data about the subject was collected from multiple sources such as industry publications, case organization's intranet websites, Internet sources, interviews and discussions with the experts in case organization, user survey and a workshop with the key users. Data was analyzed for setting the starting requirements for the case study, to gain the results of the case study implementation and to formulate the recommendations.

At the beginning, the current state analysis of the agile software development in the case organization was carried out for clarifying the environment of the case study. Current state analysis improved the understanding of the current situation and the earlier experiences gained from the subject. In addition, the current state analysis introduced the agile experiment and implementation method. The current state analysis was followed by the conceptual framework section. That offered a general level knowledge of the agile software development so that one could form a framework for researching the subject. Results of the current state analysis and the conceptual framework acted as a starting point for the case study implementation.

In the case study, the findings of the product owner's ability to carry out the case project were obtained by taking advantage of the background data and the user survey. In addition the workshop in the discovery phase of the case study provided input for the implementation. The tasks of product owner were implemented and findings of fits and caps of the tasks were collected during the case study to the Fit-GAP analysis template. Findings of each phase of the case study guided the progress of the implementation and offered a first stage information to the case organization about the product owner's ability to proceed to the next phase of the implementation.

After the case study was successfully carried out, the gained findings were analyzed and formulated to the key results. The key results were the basis for the recommendations presented in this Thesis. The aim of the recommendations was to improve the case organization's agile software development environment to support better the product owner's work. The key results and recommendations were presented to the case organization. The case organization's experts and key stakeholders offered feedback of these to the product owner.

8.2 Case Organization's Feedback

The case organization was kept up to date with the progress of the research. In addition, the findings of the case study implementation was informed to the case organization. Discussions with the case organization during the research indicated that the research was producing to them a valuable information of the product owner's role in agile software development.

The key results of the research and the recommendations formulated therefrom was presented to the case organization after the findings analyzation. When discussed with the case organization, the results were found a useful information for them to pay attention to the deficiencies of the product owner's role and organization's support. In addition, the recommendations were found useful guidance for the future development. In addition, the case organization noticed that the findings and lessons learned in this case project could be beneficial information for other product owners in the future.

The feedback from the case organization after the discussions was that the key results increased case organization's understanding of the product owner's ability to carry out the software development project in case organization's environment. Case organization had already improved the product owner's support in agile software development based to the recommendations presented in this research. As an example of that is a product owner's quick guide presented in Appendix 6 that is based on the recommendation of the handbook. Product owner' quick guide is already added as a part of the agile experiment and implementation method documentation. Other example is the product owner's network that case organization had established based on the recommendation.

8.3 Validity and Reliability

The validity in this Thesis was ensured by using the well-known methods and frameworks of agile development in the research. The data of the Thesis was collected from several sources and in multiple points during the research. Data collected from case organization's internal sources can be considered reliable. A special attention was paid to the reliability of the data collected from external sources. The external sources was used only if the data could be collected from the generally trusted sources. Avoiding the bias, one discussed many times with the people related to the agile development in the case organization and listened their opinions, presented the findings to the case organization and the key stakeholders after every phase of the case study and captured the feedback of them.

Ensuring the reliability of this Thesis, the process of data collection and analysis is described in detailed level. By that, the Thesis research can be repeated by another person. However, case projects may have different contents and scopes. That possible could lead to that the end results of the research may vary slightly.

8.4 Conclusions

The results of this research has already had an impact to the case organization's agile software development environment. However, this Thesis offers only a particular view to the subject and there would be useful to gain more information and experiences of the product owner role to achieve a wider understanding of the subject.

In addition to the product owner role, many other aspects could be researched to improve the case organization's agile development environment. During the case study one noticed some aspects that were not in the scope of this research, but are notable to consider in the future development. These are:

- Refinement of the requirements of the agile software development roles
- Increase the knowledge of agile software development in the case organization
- Conceptualize the technical support to the agile software development projects in divisions of the case organization
- Improvement of the framework agreements to support better the needs of agile software development projects
- Measurement of the maturity and quality of the agile software development
- Applying the agile experiment and implementation method to a wider field of projects, not just the software development

Aspects of the agile development listed above were presented to the case organization as a potential future development subjects.

References

- 1 City of Helsinki organization. <<https://www.hel.fi/helsinki/en/administration/administration/organization/>>. Accessed 4.1.2018.
- 2 City of Helsinki, Executive office. <<https://www.hel.fi/kanslia/en/this-is/>>. Accessed 4.1.2018
- 3 Interview, ICT development manager Ari Andersin. 15.3.2018
- 4 Interview, project manager Ilkka Kautto. 14.3.2018
- 5 AOK, Open software development unit. <<http://helmi.hel.fi/yhteisetpalvelut/tietotekniikka/kilpailutetutasantuntijapalvelut/avoinohjelmistokehitys/Sivut/default.aspx>>. Accessed 22.3.2018.
- 6 Kerppola J., Bachelor Thesis: Ketterä kehitys – case Varaamo. 2016.
- 7 Agile experiment and implementation method. <<https://digi.hel.fi/kehmet/kettera-hanke/>>. Accessed 9.2.2017.
- 8 Interview, project manager Annukka Varteva. 16.2.2018.
- 9 Interview, project manager Peter Lunberg. 13.3.2017.
- 10 Taylor C., Wilkie M., Baser J., Doing Action research: A Guide for School Support Staff. SAGE Publications. 2008.
- 11 Yin R., Case Study Research Design and Methods. SAGE Publications. 2009.
- 12 Fit-GAP analysis. <<https://www.infotivity.com/fit-gap.html>>. Accessed 14.1.2018
- 13 Agile Software development. <<https://www.agilealliance.org/agile101/>>. Accessed 18.3.2018.
- 14 Agile Manifesto. <<http://agilemanifesto.org/>>. Accessed 18.3.2018
- 15 Agile. <<https://www.atlassian.com/agile>>. Accessed 3.4.2018
- 16 Holcombe M., Running an Agile Software Development Project. John Wiley & Sons, Incorporated. 2008.
- 17 Scrum. <<https://www.atlassian.com/agile/scrum>>. Accessed 2.3.2018.
- 18 Maximini D., The Scrum Culture. Springer International Publishing Switzerland. 2015.
- 19 Scaling Agile with Scrum of Scrums. <<https://www.agilest.org/scaled-agile/scrum-of-scrums/>>. Accessed 20.3.2018.

- 20 Scrum of Scrums. <<https://www.agilealliance.org/glossary/scrum-of-scrums/>>. Accessed 3.4.2018.
- 21 The Product Owner. <<http://www.scrumguides.org/scrum-guide.html#team-po>>. Accessed 14.4.2018.
- 22 Definition of Product owner. < <https://www.agilealliance.org/glossary/product-owner>>. Accessed 14.4.2018.
- 23 GDS. <<https://gds.blog.gov.uk/about/>>. Accessed 5.2.2018.
- 24 Hinde D., PRINCE2 Study Guide. Wiley. 2012.
- 25 What is PRINCE2 <<https://www.axelos.com/best-practice-solutions/prince2/what-is-prince2>>. Accessed 10.4.2018
- 26 Coplien J., Bjørnvig G., Lean architecture: For Agile Software Development. John Wiley and & Sons, Incorporated. 2010.
- 27 Principles of Lean. <<https://www.lean.org/whatslean/principles.cfm>>. Accessed 16.2.2018.
- 28 SAFe Agile Framework <<http://www.agilest.org/scaled-agile/safe-framework/>>. Accessed 20.4.2018.
- 29 What is SAFe <<https://www.scaledagileframework.com/what-is-safe/>>. Accessed 20.4.2018.
- 30 Roles and responsibilities in projects in KEHMET. <<https://digi.hel.fi/kehmet/organisaatio/>>. Accessed 2.3.2018.

FIT-GAP Analysis Template

Nro	Phase	Task	GAP Analysis	GAP Source	GAP Requirement type	Comments

Fit-GAP Analysis of Tasks in Case Study

Nro	Phase	Task	GAP Analysis	GAP Source	GAP Requirement type	Comments
1	Project setting	Setting product owner to the project	NG			
2	Project setting	Setting the steering group to the project	NG			
3	Project setting	Defining the in-house technical support needed in the project	NG			Defined technical support for case project are AOK and Kanslia ICT
4	Project setting	Resourcing the product owner	GAP	ORG	ORG	Not following the suggestion of the experiment and implementation method
5	Project setting	Defining the key stakeholders of the project	PG	ORG	ORG	Not all stakeholders may necessarily be defined at this point. These will be completed in the upcoming phases if necessary.
6	Project setting	Product owner's familiarization to the project	PG	ORG	ORG	Not sufficient documentation or support available
7	Project setting	Project timetable	NG			Only for the experiment part
8	Project setting	Project budget	NG			Only for the experiment part
9	Project setting	Preliminary plan of the project for the steering group to get permission to start the project	NG			

10	Discovery	Defining and acquiring the organization's internal resources needed for the phase	PG	ORG	PO	Not clear understanding of expertise needed. Internal resources was completed during the phase
11	Discovery	Current state analysis	NG			
12	Discovery	User query	NG			
13	Discovery	Analyzation of the user query	NG			
14	Discovery	Committing the specialist group and the key role users to the requirement specification	PG	ORG	PO	Lack of knowledge about the specialist group and the agile way of working
15	Discovery	Service design acquisition	GAP	ORG	Other	Insufficient guidance to use the framework agreement. Needed support from the CITO
16	Discovery	Formulating the roles and requirements of the Huvaja - workshop	NG			User query, interviews and workshop
17	Discovery	Selecting the user stories for the alfa prototype implementation	NG			
18	Discovery	Options of the solution to the alfa phase	NG			In collaboration with the AOK
19	Discovery	Vision canvas	PG	MR	MR	Not information to how to use the vision canvas. Case organization could not offer support, so the project needed to acquire the support from the service design experts
20	Discovery	Preliminary roadmap	NG			
21	Discovery	Benefits comparison	NG			
22	Discovery	Phase ending presentation to the steering group	NG			
23	Alfa	Implementation plan	NG			
24	Alfa	Defining and acquiring the organization's internal resources needed for the phase	NG			Added user interface designer to the project group
25	Alfa	Using the Kanban	PG	ORG	MR	No guidance for product owner to use the method
26	Alfa	Utilizing existing data and systems	PG	MR	TECH	Not clear, if the integrations are implemented in the alfa phase. Not clear what data will be used in the prototype.

27	Alfa	Status presentation to the steering group	NG			Decision point for the implementation of the prototype
28	Alfa	Open source software coding acquisition	GAP	ORG	ORG	No guidance how to use the framework agreement or what should the invitation for tenders include. Needed support from the CITO
29	Alfa	Setting up the test group	PG	ORG	PO	Challenges in the test users timetable to have enough time for testing
30	Alfa	Service design acquisition for creating the test model	NG			
31	Alfa	Setting up the project group working and reviews	NG			
32	Alfa	Organizing the testing of the prototype	PG	Other	PO	Challenges because of the short test period
33	Alfa	Analyzation of the test feedback	NG			
34	Alfa	Analyzation of the technical solution of the prototype	NG			In collaboration with the AOK
35	Alfa	Analyzation of the information security of the prototype	NG			Prototype uses only test data
36	Alfa	Phase ending presentation to the steering group	NG			
37	Beta	Selection and prioritization of the use cases for the implementation	PG	MR	PO	Not clear understanding of the wideness of the Huvaja implementation at this point
38	Beta	Implementation plan	NG			
39	Beta	Defining and acquiring the organization's internal resources needed for the phase	PG	ORG	PO	Not clear understanding of the needed internal technical support resources. Resources was completed during the phase
40	Beta	Specifying the vision canvas	NG			
41	Beta	Specifying the roadmap	NG			
42	Beta	Selection of the architectural solutions	NG			
43	Beta	Status presentation to the steering group	NG			Actual implementation starts
44	Beta	Open source software coding acquisition	PG	ORG	ORG	Acquisition was done successfully, but there was challenges to keep in budget.

45	Beta	Building a SRUM team	PG	MR	PO	Product owner had to gain knowledge about the Scrum team building. The Scrum master was not set and the tasks of the Scrum master role had to be reorganized
46	Beta	Setting up the conditions for the definition of done	GAP	MR	PO	CITO's support was needed to define the DoD
47	Beta	Organizing the product backlog	PG	MR	PO	Product owner had not enough knowledge to do this independently. This was done in collaboration with the AOK
48	Beta	Setting up the sprint planning, reviews and retrospectives	NG			
49	Beta	Observing the development and evaluating the output according to the definition of done	NG			
50	Beta	Organizing the testing for limited group of users to test the Huvaja during the development	NG			Used the test method produced in alfa phase
51	Beta	Analyzation of the test feedback	NG			
52	Beta	Status presentation to the steering group	NG			Proceeding to the open beta testing
53	Beta	Organizing the data validation for the production-like open beta testing	PG	Other	ORG	Product owner needed a support from the Kanslia ICT to this task
54	Beta	Ensuring the support for the Huvaja during the open beta testing	NG			
55	Beta	Ensuring the support for integrations during the open beta testing	PG	Other	PO	Service vendors' contact persons for integrations support was set, but it needed a knowledge of the Kanslia ICT
56	Beta	Organizing the open beta testing	NG			
57	Beta	Analyzation of the open beta test feedback	NG			
58	Beta	Organizing the review of the information security and code quality of the Huvaja	PG	MR	TECH	Included a lack of product owner's knowledge about the case organization's manners for information security and code quality manners. This was done in collaboration with the AOK

59	Beta	Phase ending presentation to the steering group	NG			
60	Live	Transition plan from beta to live	NG			
61	Live	Defining and acquiring the organization's internal resources needed for the phase	PG	ORG	ORG	There was not set the service owner's product owner for the Huvaja. The service owner was set in the middle of the phase
62	Live	Maintenance plan of the Huvaja	NG			
63	Live	Defining the service level	NG			This was done in collaboration with the service owner
64	Live	Price inquiry for the maintaining the Huvaja	PG	MR	MR	There was not framework agreement to use for this. Guidance for producing the price inquiry was needed from the CITO
65	Live	Status presentation to the steering group	NG			
66	Live	System maintenance acquisition	NG			
67	Live	Technical review of the Huvaja	NG			
68	Live	Architectural review of the Huvaja	PG	MR	ORG	Product owner needed a support for this from the CITO
69	Live	Documentation of person register and data protection of the information system	NG			This was done in collaboration with the technical support of Kanslia ICT and AOK
70	Live	Ensuring the collection of continuous customer feedback	NG			
71	Live	Ensuring the integrations maintenance	PG	MR	TECH	
72	Live	Ensuring the organization's internal processes for Huvaja maintaining	PG	ORG	PO	The role of internal technical support after the Huvaja technical maintenance transition to the service provider was not clear.
73	Live	Continuity and recovery plans	NG			It was agreed, that the service owner's product owner will take care of there.
74	Live	Organizing the data validation	NG			
75	Live	Ensuring documentation of the information system structure, maintenance policies and agreed practices for the maintenance product owner	PG	Other	PO	There was not support for crating the documentation. Support of the CITO was needed, when producing the documentation.

76	Live	Ensuring the Huvaja technical transition to the external service provider	PG	MR	PO	Communication with the service provider and AOK was challenging
77	Live	Transferring the system form project product owner to maintenance product owner	GAP	ORG	PO	There was no model for transferring the system from one product owner to another, so there was no support for that. Product owner created the model in collaboration with the CITO
78	Live	Project ending presentation to the steering group	NG			

Vision Canvas of Huvaja in Discovery Phase

Ohjelmiston visiolakana (vision canvas)

<p>Ongelma</p> <p>LÖYTÄÄ SOPIVA TILA JA TEHDÄ VARAUS, OMA KOKOUS</p> <p>VÄHIMESTARIT: TILOJEN HOITO, KULUNVALVONTA</p>	<p>Ratkaisu</p> <p>JÄRJESTELMÄ JOLLA ETSITÄÄN JA VARATAAN TILOJA</p> <p>...JA ETSITÄÄN OSALLISTUSIA JA VARAUKSIA.</p>	<p>Ainutlaatuinen arvo</p> <p>MONOPOLI</p> <p>NOPEUS SÄÄSTÄÄ TYÖ-AIKAA</p> <p>KÄYTETTÄVYYS, LUOTETTAVUUS, NOPEUS, YLLÄPIDETTÄVYYS</p>	<p>Avainresurssit</p> <p>OHRYSSÄ HALLINTOJOHTAJA, KEHITYSPÄÄLLIKÖ</p> <p>TECH LEAD, TEKN. TOTEUTTAJAKKA LAATUKOKOUS AIEMPI KOKEMUS, KEHITYS</p> <p>PALVELUPÄÄLLIKÖ EI VIELÄ LÖYTYNYT OPEKARTIOISSA, VAIN</p> <p>PROJEKTIPÄÄLLIKÖ</p>
<p>Kenelle</p> <p>VOLYYMI PERUSKÄYTTÄJILTÄ (KANSLIAN TYÖNTEKIJÄT)</p> <p>STARPEET KUNSELTA</p> <p>VIENY TÄRIT JA VIEKÄÄN PALJON KEITTIÖ</p> <p>VARAUSUDELTA</p>	<p>Käyttäjäpalaute</p> <p>STATISTIIKKA, SPONTAANI PALAUTE</p> <p>ENSIKÄYTTÄJÄRYHYÄ, SPONTAANI PALAUTE + TESTIT</p> <p>PALAUTENAPPI</p>	<p>Rajoitukset</p> <p>VARAUSTEN SIIRYTTÄVÄ VUOSI ETEENPÄIN</p> <p>SELAINPOHJAINEN</p> <p>ILMISTEN AIKATAULU-AVUSTA JA AJANKOHTAISUUS</p>	<p>TILANVARAUKSEN TOIMITTAVAIN JÄTEKUVASTI, VAIN LYHYITÄ KATEOJA</p> <p>TAVOITE-AIKATAULU 2017</p> <p>ESIM. Uudet tilat siirtyvät ensin</p>
<p>Tekninen visio</p> <p>JATKUKVA KEHITYS HALLITUSTI, ERIYTETIT MODULIT</p> <p>HYÖDYNNETÄÄN MUUN KAUPUNGIN JÄRJESTELMIEN OPIT</p> <p>TUKEE TALOUDEN JÄRJESTELMIÄ</p>	<p>EROON PALMIAN VIRASTOKONEESTA</p> <p>VARAAMON KEHITYKSEN HYÖDYNTÄMINEN</p> <p>AVOIMET RAJAPINNAT</p> <p>RESPA POHJAKSI, VARAAMO + OMAA KOODIA, PÄÄTÄLI KÄYTTÖLIT</p> <p>EXCHANGE-INTEGROATIO</p> <p>TUNNISTAUTUMINEN, POOLITTA AD:TA VASTEN</p>	<p>Liiketoimintamittarit</p> <p>LASKELMAT TYÖAJAN SÄÄSTÖSTÄ PER POOLI</p> <p>PERUSKÄYTTÄJÄN TYYTYVÄISYYS</p>	<p>ALFA BUDJETTI SO KE, BUDJETOIDUN HANKE-RAHAN RIITÄVYYS</p>

Test Model

Huonevarausjärjestelmän testaus

Palautelomake

Vastatessa on hyvä muistaa, että nyt testattu versio on keskeneräinen ja siitä saattaa puuttua vielä ominaisuuksia. Palautetta pyydetäänkin liittyen nyt testattuihin ominaisuuksiin sekä testaukseen perustuvaan mielikuvaan uudesta lopullisesta huonevarausjärjestelmästä.

Mikä on yleis mielikuvasi järjestelmän uudistuksesta?

0 1 2 3 4 5

Mikä on yleis mielikuvasi järjestelmän käytettävyydestä?

0 1 2 3 4 5

Mikä on yleis mielikuvasi järjestelmän visuaalisuudesta?

0 1 2 3 4 5

Miten paljon oletat uuden järjestelmän helpottavan työtäsi?

0 1 2 3 4 5

Miten paljon oletat uuden järjestelmän säästävän työaikaasi?

0 1 2 3 4 5

Miten todennäköisesti suosittelet uutta järjestelmää työkaverillesi?

0 1 2 3 4 5

Muita kommentteja ja ajatuksia huonevarausjärjestelmän kehityksestä? Sana on vapaa!

Kiitos osallistumisestasi!

Huonevarausjärjestelmän testaus

Saatko järjestelmällä halutut asiat tehtyä?

Onko käyttöliittymä looginen ja toimiva?

Ovatko visuaaliset elementit ymmärrettäviä?

Tuleeko käytössä vastaan ongelmatilanteita?

Miten muuttaisit tai parantaisit järjestelmää?

Miten järjestelmä palvelisi paremmin juuri sinua?

Onko järjestelmässä itsellesi ylimääräisiä ominaisuuksia?

Puuttuuko järjestelmästä jotain?

Muita kommentteja?

Kiltos osallistumisestasi!

Checklist for Transition

Tuotantoon siirtyminen - muistilista

Tuoteomistajan näkökulma

- Järjestelmän toiminnallinen kuvaus
 - Mikä järjestelmä on ja mitä se tuottaa
- Järjestelmän tekninen kuvaus (sillä tasolla, että tuoteomistaja ymmärtää, mistä on kyse)
- Backlog
 - Tarvittaessa kuvaus kehityksen nykytilasta ja avoinna olevista asioista
- Kuvaus sovelluksen toiminnallisesta arkkitehtuurista, integraatioista ja niiden hallintaprosesseista
 - Esim. jos tulee ongelma, mistä lähdetään kysymään ja selvittämään
- Kuvaus siitä, miten ylläpito on järjestetty, millainen sopimus siitä on tehty
 - Mitä sopimus kattaa (ongelmanselvitys, virheenkorjaukset jne.)
 - Millainen palvelulupaus, SLA (palvelutaso, palveluajat jne.)
- Kuvaus järjestelmän ylläpitoon liittyvistä rooleista ja käyttöoikeuksista
 - Ylläpitoon liittyvät roolit (tuoteomistaja, pääkäyttäjä, sisällöntuottaja jne.)
 - Järjestelmän käyttöön liittyvät roolit (pääkäyttäjä, super user, peruskäyttäjä jne.)
 - Käyttöoikeudet (muokkausoikeudet, katseluoikeudet)
- Kuvaus käytössä olevat prosesseista ja työkaluista (esim. uuden käyttöoikeuden lisääminen, sisällön lisääminen järjestelmään, uusien ominaisuuksien käyttöönotto)
 - Kuka tekee, mitä tekee, tehdäänkö itse vai teetetäänkö
- Kuvaus yhteistyötahoista ja sidosryhmistä sekä siitä, mitä on sovittu kommunikaatiosta näiden kanssa
 - Sovitut yhteistyötavat ja kommunikaatiokanavat
- Kuvaus kehityshankkeista, jotka mahdollisesti vaikuttavat järjestelmään ja palveluun
- Kuvaus sovituista jatkokehitys- ja yhteistyönäkymistä

Tekninen kuvaus

- Saatavana tuoteomistajalle sillä tasolla, että hän tietää, mitä asioita tekniseen kuvaukseen sisältyy
- Arkkitehtuuri
 - Riittävän tason kuvaus tuotteen toiminnallisesta arkkitehtuurista ja integraatioista teknisellä tasolla
 - Kuvaus arkkitehtuuripoikkeamista yksityiskohtaisella tasolla
 - Kuva suositeltava
 - Laaditaan yhdessä ka-asiantuntijan kanssa
 - miten tuote sijoittuu kaupungin kokonaisarkkitehtuuriin
- Tietoturva
 - Palvelukohtainen tietoturva
 - Riskiarvio: minkätasosta tietoturvaa tarvitaan?
 - Tarvitaanko auditointia? Jos tarvitaan, minkätasosta auditointia?
 - Kuvaus siitä, miten tiedot on turvattu teknisesti
 - Esim. turvataanko tiedot jo alustassa vai onko toteutettu omia tietoturvaprotokollia
 - Onko käytetty kirjastoja, joiden oma ylläpitoprosessi saattaa olla puutteellinen, ja miksi?

- Tietosuojaja
 - Onko tietosuojan vaatimat tekniset valmiudet toteutettu?
 - oikeus tulla unohdetuksi
 - oikeus korjata omat tietonsa
 - oikeus siirtää omat tietonsa
 - oikeus tietää, miten omia tietoja käsitellään
 - mahdollisuus todentaa, kuka on katsonut tietoja
 - ks. GDPR tai kysy Villeltä
- Tekninen luotettavuusarvio palvelulupausta varten
 - Miten asennettu, miten koodattu, mikä
 - Mikä on tuotteen eri osa-alueiden luotettavuus (esim. integraatiot)?
 - Testauksen taso
 - Onko kuormitustestaus tehty?
- Tarvittavat automaatiot
 - Tuotannon aikainen valvonta
 - Tietojen varmistus
 - Tiedon varmistusten taso
 - Palautussuunnitelma
 - Tiedon tallennustavan vaikutus varmistuksiin, esim. versiointi, auditointi

Product owner' quick guide

Tuoteomistajan pikastartti

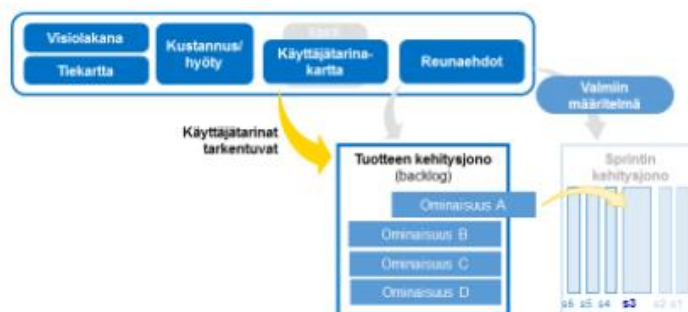
Tuoteomistaja toimii palvelun omistajan ohjauksessa. Hänen tehtävänä on pitää tuotteen ylläpitoon ja kehitykseen liittyvä kokonaiskuva hallussa koko ajan. Tuoteomistaja huolehtii, että kehityksessä toteutetaan ohjausryhmän hyväksymää kokonaistavoitetta. Hän huolehtii, että käyttäjien tarpeet on määritelty oikein, että lopputulos on näiden vaatimusten mukainen ja että tuotantokäytössä täyttyy palvelulta odotettu palvelutaso. Ketterässä hankkeessa tuoteomistaja vastaa tuotteen kehitysjonon priorisoinnista.

Usein tuoteomistajan rooliin yhdistyy myös projektipäällikön vastuita, etenkin silloin kun tuotteessa on meneillään kehitysvaihe. Käytännössä tuoteomistaja on myös tuotteen ympärille syntyneen toimijaverkoston puheenjohtaja.

On huomattava, että tuotteenhallinta säilyy tuoteomistajalla, vaikka tuotteen ylläpito olisi ulkoistettu.

Ks. myös <https://digi.hel.fi/kehmet/sanasto---uusi/tarkeimmat-kasitteet>

Alla oleva kuva havainnollistaa tuoteomistajan vastuita. Tummalla merkityt osat ovat tuoteomistajan vastuulla:



(Alkuperäinen kuva: <https://digi.hel.fi/kehmet/metelmalaari/betan-valmistelu>)

Kaupunki- tai toimialatasoinen yhdenmukaisuus

Tuoteomistaja huolehtii, että ratkaisu kiinnitetään kaupunkitasoiseen kokonaisarkkitehtuuriin. Huomioon otettavia asioita ovat esimerkiksi:

- Palvelumuotoilu
- Rajapinnat
- Teknologiat
- Tietovarannot
- Järjestelmäarkkitehtuuri
- Toimintakuvaukset (prosessit)

Prosessin hallinta

- Huolehtii siitä, että palveluun tarvittavat avainroolit määritellään ja täytetään sekä ohjaa avainhenkilöiden yhteistyötä (esim. pääkäyttäjät, sisällöntuottajat, käyttäjät)
- Hallitsee operatiivista prosessia
 - Pitkän tähtäimen suunnitelma
 - Budjetointi
 - Säännölliset statuspalaverit sidosryhmien kanssa
 - Ohjausryhmän koolle kutsuminen ja kokousten valmistelu
 - Ylätason aikataulujen ja määräaikojen hallinta ennakoivasti
- Vastaa versionhallinnasta ja sen prosessista ja projektoinnista
- Pitää kirjaa sovituista asioista
- Pitää huolen, että tuotteelle laaditaan tarpeenmukainen dokumentaatio

Budjetin hallinta

- Huolehtii, että budjetinkäyttö on suunnitelmallista
- Seuraa budjetin käyttöä
- Raportoi mahdollisista poikkeamista ohjausryhmälle ennakoivasti

Resurssien hallinta (ulkoa ostetuissa projekteissa)

- Huolehtii kilpailutuksesta tai puitesopimusten käytöstä yhdessä osaamiskeskuksen ja avoimen ohjelmistokehityksen asiantuntijaryhmän kanssa (silloin kun käytetään avoimen ohjelmistokehityksen puitesopimusta)
- Ohjaa ulkoa ostetun tiimin toimintaa (mahdollisesti yhdessä oman tech leadin kanssa)

Tuotteen kehitysjonon (backlog) hallinta

- Ylläpitää tuotteen kehitysjonoa (kokoaa kehitystarpeet ja muutosehdotukset keskitetysti)
- Asettaa kehitystarpeet tärkeysjärjestykseen

Muutoksenhallinta

- Huolehtii muutoksenhallinnasta: Jos kokonaistavoite muuttuu olennaisesti, tuoteomistaja huolehtii siitä, että muutokset sekä niiden aiheuttamat muutokset budjettiin ja kokonaisaikatauluun käsitellään ohjausryhmässä ja viedään työjonoon.
- Huomaa, että muutoksenhallinta tarkoittaa perinteisessä ja ketterässä lähestymistavassa hieman eri asioita:
<https://digi.hel.fi/kehmet/menetelmalaari/muutosten-hallinta>

Viestintä ja vuorovaikutus

Tuoteomistaja huolehtii, että ratkaisusta ja siihen liittyvistä asioista käydään vuoropuhelua tarvittavien tahojen kanssa.

- Kommunikoi ylläpidosta ulkoistetun palveluntuottajan kanssa
- Kommunikoi projektin ohjausryhmään asioiden etenemisestä
- Kommunikoi muiden sidosryhmien kanssa
- Käy jatkuvaa vuoropuhelua sisäisten sidosryhmien kanssa (yhtymäkohdat muihin kehitysprojekteihin)

- Käy jatkuvaa vuoropuhelua niiden palvelujen edustajien kanssa, joita tarvitaan oman palvelun tuottamiseksi (esim. integraatiot, tunnistautuminen jne.)
- Viestii sidosryhmille aktiivisesti versiomuutoksista
- Jos tuotteella on useita instansseja ja käyttäjiä eri organisaatioissa: pitää yllä verkostoa ja tekee yhteistyötä muiden saman tuotteen käyttäjien kanssa sekä ylläpitää tietoa muista käyttäjistä (esim. muut toimialat, muut kaupungit, yhteiskehityksen koordinaattori)
- Ohjaa sisältöön liittyvät palautteet oikeaan osoitteeseen

Vastuunjakotaulukko

Tuoteomistaja päättää, mitä tehdään.

Tech lead päättää, miten tehdään.

Tehtävä	Vastuuhenkilö: tuoteomistaja	Vastuuhenkilö: tech lead
Kokonaiskuvan hallinta	x	
Budjetin hallinta	x	
Kehityssykljen hallinta	x	
Tietoarkkitehtuurin hallinta	x	
Jatkuvien palvelujen hallinta	x	
Ylläpidon toimittajahallinta	x	
Tuotantoympäristön hallinta	x	
Integraatioiden hallinta	x	
Kehitysresurssien hankinta ja ohjaaminen (ulkoa ostetut resurssit)	x	
Tuotteen kehitysjonon hallinta	x	
Sprintin suunnittelu	x	
Sprintin kehitysjonon hallinta		x

Hyväksymistestaukset, käyttäjättestaukset	x
Teknisen kehityksen ohjaaminen	x
Teknologioiden valitseminen	x
Tekninen testaaminen	x
Laadunvarmistus	x
Koodin katselmointi	x
Teknisten ongelmien selvittäminen ja ratkaiseminen	x
Tietolähteiden virheiden havainnointi	x