Tampere University

Ville-Valtteri Jäävalli

Applying agile to portfolio management

Faculty of Information Technology and Communication Sciences (ITC)
Master of Science thesis
November 2019

ABSTRACT

VILLE-VALTTERI JÄÄVALLI: Applying agile to portfolio management
Tampere University
Master of Science thesis, 45 pages
November 2019
Master's Degree Programme in Information Technology
Major: Software engineering
Examiner: Prof. Kari Systa and Samuli Pekkola
Keywords: Agile, portfolio management

Portfolio management is a practice of managing organization's projects in a structured manner. The goal is to produce maximum value by selecting executed projects, realizing the maximum amount of business value. Portfolio management is also one of the most important tools to execute company strategy by focusing on development efforts and as such an important tool for management. Agile project management and development practices have become very common in the last decade, so portfolio management should also be compatible with modern agile development methods when applied.

In this thesis, a literature review is made to understand the current research of portfolio management, agile development methods, and combining those two subjects. A goal is to understand how portfolio management process should be defined to be compatible with projects using agile development methods, and if agile methods can be applied in the process as well.

A new customized portfolio management framework is designed and implemented based on the literature review for Elo Mutual Pension Insurance Company's investment organization to be used to manage in house development projects. Thesis will follow up if the framework is able to improve organization's portfolio management practices.

The designed framework was considered as an improvement and was designed to support use of agile development methods.

TIIVISTELMÄ

VILLE-VALTTERI JÄÄVALLI: Applying agile to portfolio management
Tampereen Yliopisto
Diplomityö, 45 sivua
Lokakuu 2019
Tietotekniikan koulutusohjelma
Pääaine: Ohjelmistotuotanto
Tarkastaja: Professori Kari Systa and Samuli Pekkola
Avainsanat: Ketterä, portfolion hallinta

Portfolion hallinta tarkoittaa organisaation projektien hallinnointia järjestelmällisellä tavalla. Tarkoituksena on luoda mahdollisimman paljon liiketoiminta-arvoa valitsemalla parhaiten arvoa tuottavat ja yhtiön strategiaan sopivat projektit, sekä hallinnoida näitä tehokkaasti. Kehityspanostusten ohjaaminen portfolion hallinnan kautta on tapa toteuttaa yhtiön strategiaa ja täten tärkeä työkalu johdolle. Ketterät kehitysmenetelmät ovat yleistyneet viime vuosikymmenellä, joten portfolion hallinnan prosessin tulee myös olla yhteensopiva näiden kanssa niitä sovellettaessa.

Tässä työssä tehdään katsaus nykyiseen portfolion hallintaa, ketteriä menetelmiä ja ketterää portfolion hallintaa käsittelevään kirjallisuuteen ja tutkimukseen. Tavoitteena ymmärtää miten portfolion hallintaprosessi tulee toteuttaa, jotta se toimii projektitasolla sovellettavien ketterien kehitysmenetelmien kanssa, sekä tutkia miten ketterän kehityksen periaatteita voidaan soveltaa itse prosessiin. Löydöksien perusteella työssä suunnitellaan ja kuvataan Työeläkeyhtiö Elon sijoitusorganisaatiolle luotu portfolion hallinnan malli.

Diplomityö seuraa uuden portfolion hallinnan mallin käyttöönottoa Elolla neljän kuukauden ajan ja seuraa millaisia vaikutuksia sen käyttöönotolla on organisaation sisäiseen kehitystyöhön. Toteutuksen toimivuutta, hyötyjä ja haasteita arvioidaan kvalitatiivisia mittareita, kuten palautetta hyödyntäen.

Luodun mallin avulla pystyttiin parantamaan Elon sijoitusorganisaation portfolion hallintaa ja aiheuttaman kehitystyöhön positiivisia vaikutuksia. Kirjallisuudesta onnistuttiin tunnistamaan olennaisia ratkaistavia ongelmia portfolion hallinnan ja ketterän kehityksen yhdistämiseksi, jotka otettiin huomioon prosessien suunnittelussa.

PREFACE

I started working in Elo mutual insurance pension fund in autumn 2018. I did know that Elo had some ambitions towards developing ways of working, but I had no idea how large part of my daily work this subject would eventually form. A project to design a new development framework was started before I even began my Elo career, but eventually, I was appointed to be a project manager for this project during October. This master's thesis is done as a part of this project.

Before I was involved with this project, I was a bit familiar with agile development methods as I have a background from software development. In retrospect, my knowledge was quite shallow. During the project, I learned a huge amount about agile development methods and general project portfolio management.

I would like to thank everyone who participated in the project. Teemu Eskelinen, Nina Nieminen, Saku Sairanen, Kati Saarni, and Kati Reinikainen. Also, thanks to Jarno Vähäniitty and Pasi Rupponen, who both provided valuable insight into agile methods during the project.

Ville-Valtteri Jäävalli
Helsinki, November 2019

CONTENTS

1. INTRODUCTION

This chapter will describe the background for this thesis. The scope, research questions and goals for this study are introduced.

1.1 Background and motivation

On Autumn 2018, the management of Elo Mutual Pension Insurance Company's investment organization had identified they had issues of controlling ongoing development activities. A project was started to resolve these issues. The goal of the project was to create a framework to manage development activities across investment organization. This was expected to improve the ability to be adaptive and efficient. These capabilities are generally identified as crucial factors for an enterprise to survive[1].

Before the project was started, it had been identified that there were some issues in communication between different teams and working units. Prioritization of development activities was not clear, and visibility to resource utilization was limited. An old formal framework to manage project portfolio existed, but it was experienced as too burdensome on bureaucracy as it was focused on large scale projects including external consultants and operated on the scope of the whole organization. It was considered not suitable for managing lighter in house development work inside investment organization. As a result, many development activities were pushed forward within different teams without any management framework, and information was not shared properly. Different groups often ended up competing shared resources as there was no common understanding about priorities, which caused confusion and challenges to focus resources towards development activities generating the largest amount of business value.

Agile development methods were not commonly used, but a small team of analysts had been experimenting with agile development during an internal software development project and had promising results. Based on their experiences and the vision of the management, it was decided that the portfolio management framework should be compatible with an agile way of working. A team was assembled to design and

implement processes needed to form a complete portfolio management framework to manage internal development activities and to resolve identified issues. This study is made as a part of the project.

A large amount of literature and research about portfolio management is available, but in general, the focus is on the new product development. The subject of agile development is also researched but generally in the context of software development. There is also some research about combining these two subjects, which is referred to as agile portfolio management in the literature, but once again generally in the context of software development and only limited amount of empirical evidence can be found[2]. Even though software development projects are also relevant in the context of Elo Mutual Pension Insurance Company's investment organization, the majority of development activities are related to improving operative efficiency or increasing capabilities to do analysis work to improve quality of investment decisions. So the challenge is to apply what can be learned from the literature in a different context.

1.2 Scope

Even though the project included a fair share of research about how development teams should work in practice, this is left out of the scope. The focus is to find out how to manage development activities efficiently and support agile development methods in the context of Elo Mutual Pension Insurance Company's investment organization. Some project management literature is introduced to understand how to integrate project management practices into the portfolio management process.

This study won't address the issues of working with external parties as most development work is done in house without or with only a limited amount of involvement from external parties.

It should be noted that this study includes only Elo Mutual Pension Insurance Company's investment organization, which is only a part of the company. It involves around 100 employees. Elo Mutual Pension Insurance Company has approximately 500 employees in total as of 2019.

1.3 Goal

The first goal of this study is to find out how portfolio management should be done in such a way that it supports agile development methods and how agile principles

can be applied to a portfolio management process as well. This is studied as a literature review.

A new portfolio management framework is designed and implemented based on the literature reviewed and requirements set by the organization. This study will followup how it is received and evaluate the impact on managing development activities using qualitative measures such as feedback from the management and employees.

1.4 Research questions and plan

The questions this thesis aims to find answers are:

- RQ1: What adjustments to portfolio management should be made to make it compatible with agile development methods?
- RQ2: What agile principles could be applied in context of Elo Mutual Pension Insurance Company's investment organization to increase agility in the portfolio management process?
- RQ2: What are the first actions that should be taken in effort to introduce portfolio management to produce maximum business value?

Answers for the questions are searched from the literature and based on findings after a new framework has been implemented.

1.5 Methodology

This study follows design science principles. Peffers, Tuunanen, Rothenberg and Chatterjee have described following activities[3] that represent different stages of research:

1. Problem identification and motivation
2. Define the objectives for a solution
3. Design and development
4. Demonstration
5. Evaluation

6. Communication

Problem identification, motivation, and objectives are described on the chapter 1. The design and development of the framework are introduced in chapter 3 based on the research in the chapter 2. Results and evaluation are presented in the chapter 5.

1.6 Project description

The project initially started during summer 2018, but the author joined it in on autumn. The design of the framework was done between September and December 2018. Introducing the framework to the organization was done in January 2019, and the framework has been used since that.

The goals of the project are partly similar to the ones this thesis has. But of course, results matter, so for the project to be considered successful, it has to be able to improve the development portfolio management process, raise awareness about ongoing development projects, increase capabilities and introduce these new practices properly. This includes creating a suitable customized portfolio management framework, training participants, facilitating needed ceremonies, and other tasks related to implementation.

2. LITERATURE REVIEW

This literature review includes parts from the portfolio management, requirements management, and agile development literature. The goal is to find out how portfolio management should be practiced with the use of agile development methods and if agile principles can be applied to the portfolio management process itself. The literature separates program management from portfolio management, but there is significant overlap as both addresses the issue of managing development activities[2], so this chapter includes reviews from both areas of literature.

2.1 Portfolio management

In a context of finance portfolio management is commonly used to refer to managing portfolios of investments, but in this study, it is used to refer to managing portfolios of different development activities. According to Shan Rajegopal, Philip McGuin and James Waller terminology for project portfolio management are indeed borrowed from the investment community[4], which feels natural as both topics share some same elements like managing risk and balancing portfolio to meet organizations strategic goals.

A **portfolio** is defined by Project Management Institute as a collection of projects or programs that are grouped together for efficient management[5]. Going forward, Cooper defines **project portfolio management** as a dynamic process where all of the company's development and new product projects are revised, evaluated, prioritized, and selected for execution. Old projects can be killed or prioritized and resources re-allocated[6]. He also defined that the goal is to produce the maximum amount of value for an organization, to balance between different parameters like risk versus profit, and ensure portfolio is reflecting company strategic direction. Decisions about product lifecycle, development strategy and establishing partnerships, are also part of the process[7].

In a broader context, portfolio management is seen as a part of **new product development (NPD)**, which is an area of management literature. It studies the overall process of how new products should be delivered from concept to production,

including strategic decision making, product planning processes, and commercialization[8]. The term **portfolio management of new products** is often used in the literature, but this thesis uses the term portfolio management to describe the process.

As portfolio management links directly to strategy and efficient resource allocation, it is a vital issue for any company[9]. It is a tool to maximize produced value, so it comes with no surprise that portfolio management is viewed as a very important task by senior management[6]. However, the portfolio management process might not be as necessary for small organizations as it is for a large organization. Vähäniitty studied portfolio management in agile environments and concluded that small organizations would probably suffer from a lack of portfolio management in certain conditions, but it might not cause problems[10]. This is because portfolio management decisions are made anyway, regardless of the actual decision-making process, and the amount of ongoing development activities may be simple enough to handle without a formal process. Still, he identified portfolio management along with a process called roadmapping as the most crucial part in connecting business and development decision making.

A portfolio contains development activities that need to be structured in a meaningful manner. The next subsections describe some concepts and processes that are generally used to manage development activities inside a portfolio.

2.1.1 Programs

A program is a collection of related projects that are managed together because it delivers additional value compared to just managing each project independently[5]. Projects inside programs might pursue the same strategic goal or have some other joint outcome.

Program management has its own organizational structure and roles. Roles are including a sponsor from senior management responsible for investment decisions and a program manager responsible for managing a program and realizing its benefits[11]. Even though there is much literature about programs as a general method to organize development, there has been a study indicating that much of the assumed benefits are not realizing[12]. Some reasons mentioned are managements excessive control focus causing bureaucracy and failure to facilitate genuine co-operation between project managers.

2.1.2 Project management

Regardless of how projects are managed on the top level, there is a need to manage each individual project somehow. Using definitions provided by Project Management Institute **project management** is a practice of applying knowledge, skills, tools, and techniques to project activities to meet the project requirements. It involves five stages defined in the figure 2.1.[5]

Figure 2.1 *Phases in project management*

Managing a project includes identifying requirements, communicating with stakeholders, and balancing with constraints such as resources, quality, scope, schedule budget, and risk. However, there is no way to define an ideal structure for a single project, so in practice, many different management models are used.[5]

Understanding a company's project management is important in order to integrate it with portfolio management processes. Section 2.4 researches how to integrate these processes while using agile development methods.

2.1.3 Stage-gate management model

Already in 1986, Cooper defined the stage-gate model in his book as a conceptual and operational model for moving new product ideas to launch[13]. Each stage includes multiple predefined parallel activities, and gates are control points for decision making. So before entering a stage decision has to be made at the gate.

Multiple variations of the stage-gate model can be found from the literature like the classic Cooper stage-gate model[14] and Mulders PROPS model[15]. It also has a commercialized[16] version. One of the stage-gate models described by Cooper[17] includes stages described in the table 2.1.

Gates represent decision points between stages. Gates have inputs, which are the results from the previous stage, criteria for making the actual decision, and outputs that will be deliverables for the next stage. The figure 2.2 illustrates example of a decision workflow in a gate.

The decision options at the gate are to:

Stage	Description
Preliminary Investigation	A quick investigation and scoping
Detailed Investigation	Creating an actual business case. Market research, financial analysis, technical assessment etc necessary research of relevant considerations. Also planning action for next steps.
Development	The implementation phase. Project is executed based on plans and research from previous step.
Testing and validation	The verification and validating the results from development.
Full production and market launch	Commercialization of the product.

Table 2.1 *Stages in Stage-gate model*

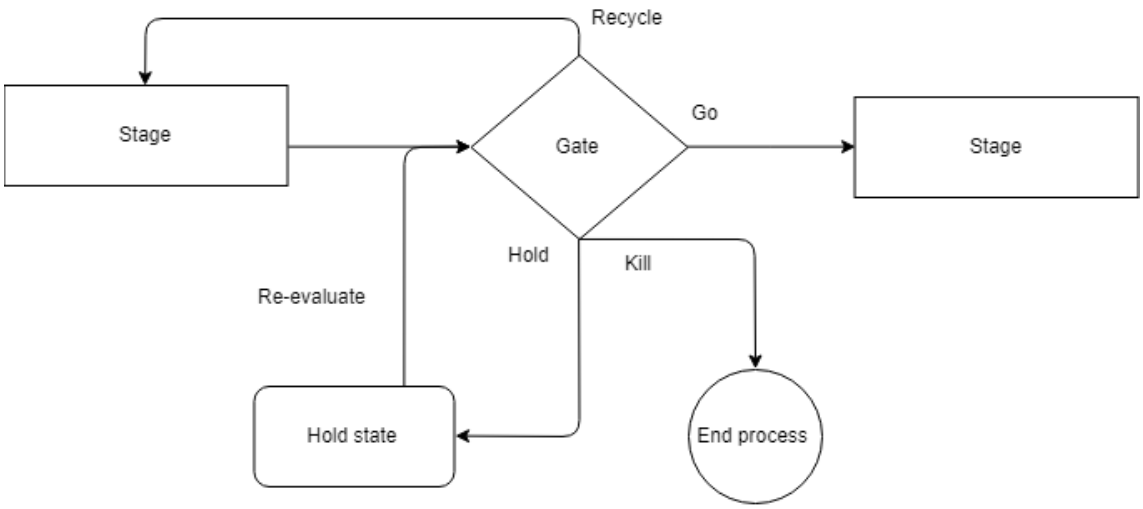

Figure 2.2 *Decision diagram for stage gate*

- Go: Proceed to next stage
- Kill: Project is terminated
- Hold: Project will be on hold until the decision to continue will be made.
- Recycle: Return project to previous stage,

When comparing Cooper's stages with phases used by Project Management Institute, they both share a quite similar structure, but Cooper also introduces a structured decision-making process to control different phases.

2.1.4 Project management office PMO

A commonly used way to organize and support projects is to have a separate project management office (PMO), which is a business unit dedicated to improving the practice and the results of project management[18]. According to Kendall, the role of the PMO can be mentoring and focused on establishing processes and supporting, or it can provide project managers as resources to various projects. So the actual role of PMO can vary between different organizations.

According to Project Management Institute[5], responsibilities of PMO may include the following:

- Managing shared resources
- Identifying and developing project management methodology, best practices, and standards
- Coaching, mentoring, training, oversight
- Developing and managing policies
- Monitoring compliance
- Coordinating communication across projects

So the role of PMO is usually to help manage project portfolio and support executing projects efficiently, but priorities and the actual assignments come from the business management. Hodggings and Hohmann suggest that PMO is essential, especially when dealing with related product offerings from a single business unit, which can lead to competing for limited resources[19]. They researched PMO in an organization that was using Scrum as a development method and concluded that adoption of agile development methods the PMO change and be fully integrated with development processes.

2.1.5 Relations between portfolio, programs and projects

There is an unlimited number of ways to structure development work in an organization, and suitable structures to govern vary by organization. Program management institute defines relationships as different layers of governance. A project is governed

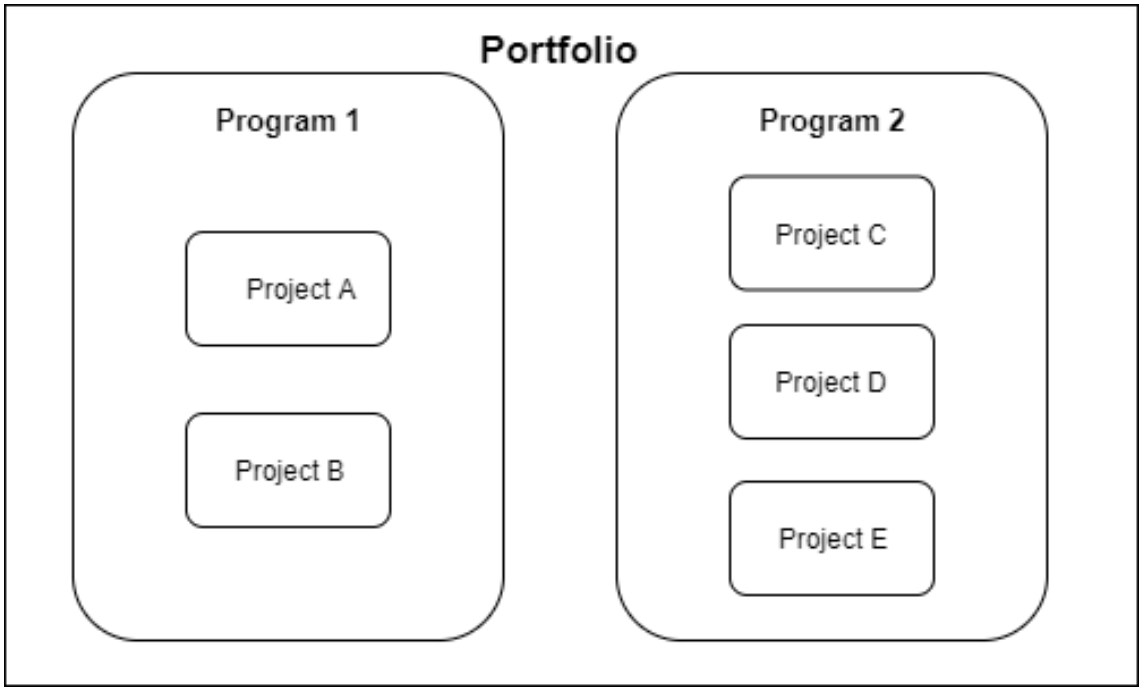

Figure 2.3 *Relations in portfolio management*

inside the project management process. Multiple projects are governed inside programs and programs are governed in a portfolio[5]. These relations are illustrated in the figure 2.3.

In practice, organizations can have many variations of this structure. For example, there can be multiple different portfolios that can be grouped inside another portfolio, which might be viewed by top management. On the other hand, the concept of program or even portfolio might not be used at all.

2.1.6 Approaches to project selection

There are many different quantitative and qualitative approaches that can be used to choose projects that should be selected for development inside the portfolio. Companies can make use of one or more of these approaches at the same time. Research suggests that most successful companies are utilizing more than one approach, and companies relying on only a single approach tend to perform poorly, so it might be best to apply more than one method.

The most obvious and commonly used one is to evaluate projects using financial methods like using net present value valuations or discounted cash flows. But as managers responsible for the projects are often under pressure to explain the value for senior management, the estimates tend to be biassed towards too optimistic num-

bers[20]. Financial calculations also include a significant amount of uncertainty in general because they are more or less based on assumptions or sometimes can not be estimated at all. Cooper warns about relying too much on financial approaches, as research suggests that companies relying primary on financial methods tend to form poorly performing project portfolios[6]. This is in line with a case study conducted by Kester, Hultink, Erik, and Lauche as they concluded that companies using only quantitative measures have a challenge of missing innovation opportunities[21]. According to this study, a company that recognizes that projects of different levels of innovativeness require different evaluation approaches most likely will have success in the long run.

Another popular approach is to select projects according to business strategy. Money is allocated to projects reflecting strategic decisions made by the management[22].

The agile portfolio management literature suggests the use of different visualizations to identify high-value projects.[23]. A simple example is to create a graph with a probability of success on the y-axis and value on the x-axis. Then projects can be compared relative to each other and pick projects with the best risk versus value ratios. The problem is that those metrics need to estimated somehow, which might be difficult.

A one way to systematically evaluate projects is to create a scoring model that produces numerical measures to estimate overall value. The model can include multiple qualitative and quantitative questions. The results can then be used to prioritize projects among each other. The disadvantage of this approach is that it can be hard to compare attributes between each other, especially if projects vary in the size of the scope effort significantly.[22][6]

In addition to the already mentioned methods, many more complex suggestions can be found in the literature. For example, there are studies applying option theory for portfolio selection[24][22]. The idea is to treat each stage of a project as an option for future investment. Other mathematical models, such as using fuzzy triangular numbers and fuzzy programming by Nancy M. Arratia-Martinez[25] are suggested, but it seems there is no literature indicating those are actually used in business environments.

2.1.7 Measuring portfolio performance

Measuring the performance of a development portfolio can be difficult to express as quantitative metrics, but it can be done with qualitative methods by interviewing

the management. Cooper defines six performance goals that portfolio management should follow[22]:

- Having right number of projects in the portfolio compared to resources
- Undertaking projects on time an in a time-efficient manner
- Having high value projects in the portfolio
- Balanced portfolio. Long term risk and profits versus short term etc
- A portfolio is aligned with strategy
- Spending breakdown of portfolio mirrors the business strategy and priorities

The velocity of development work can be measured using quantitative methods if the estimation of work has been standardized[26], but it won't reveal if development efforts are allocated to relevant business goals. The agile portfolio management literature seems to focus on finding metrics to measure how projects are performing and not so much from the strategic perspective[23]. However, according to Martinsuo and Lehtonen, portfolio management efficiency is directly linked to project management efficiency[27]. So in order to achieve a well-performing portfolio, a company should have efficient project management practices.

2.1.8 Common challenges

It might seem quite easy to create a rational and efficient project portfolio management framework, but real life can be much more complex than theory. As Martinsuo discovered, decision making is much less rational than the normative models would suggest[28]. Situations are different, so portfolio management needs to be applied differently, adjusting to the situation. Parameters like risk, priority, and business value are quite ambiguous, so it follows that portfolio management can neither be unambiguous. Even though it is possible to create highly sophisticated models, the quality of data for the inputs might not be good enough to make results accurate and useful[9][22].

Even though it is hard to create a good theoretical model to manage a portfolio that is also working in a real-life, the subject should not be ignored. Cooper interviewed managers and found out that poor portfolio management can lead to several serious negative consequences[6]. These consequences are including executing low-value projects, lack of focus and missing strategic criteria.

2.2 Requirements management

Requirement is defined by Project Management Institute as "*a condition or capability that is required to be present in a product, service, or result to satisfy a contract or other formally imposed specification*"[29]. Requirements can be user scenarios, functions, feature lists, analysis models, or specifications[30]. All development activities are composed of different requirements. On a top-level, they can be defined either as performance requirements, which defines what must be able to perform or design constraints which describes boundaries that should be followed to complete performance requirements[31]. Requirements management is a broad subject, so the focus will be on abstraction models and backlogs as they are a relevant part of the portfolio management process[32] and suggested to be used by the agile development literature[26].

2.2.1 Backlogs

A backlog or product backlog is a set of work items and is used to coordinate work that should be done. Its main function is to communicate the desired order, not to act as detailed specifications[33].

Generating a backlog is a result of process called a **grooming** in the agile literature. This process includes creating new work items into a backlog, reassessing priorities, removing stories that are not relevant, and assigning estimates[34][33]. Prioritization is covered in section 2.2.3. There are many different techniques to assign estimates, like estimating ideal time. Table 2.2 describes a few estimation techniques from a book written by Ashmore and Runyan[35].

Technique	Description
T-Shirt sizing	Simple rough categorisation. For example small, medium, and large.
Ideal time	Estimation about time required in ideal conditions.
Hours	Simple hour estimation.
Story points	An arbitrary measure to understand the size of the effort.

***Table 2.2** Different estimation techniques*

To create an actual estimate, a team can use methods like planning poker where all team members do an estimate using selected measure and estimation is made based on the results[35].

Backlogs are an important concept in agile development frameworks such as Scrum, which is the most used agile framework[23]. However, backlogs that are focused on

project or product levels are not enough for addressing the business needs of the organization. Hodgings and Hohmann implemented business unit level backlogs to support the integration of business requirements with the development process[19]. Use of strategic backlogs is also highlighted by Stettina and Hörz[2].

2.2.2 Abstraction models for requirements

Requirements have varying scopes and abstractions. An abstraction model is needed to make different sizes of scopes and work included comparable with each other, and it helps management to understand better what should be developed. This also allows prioritization and planning development activities on many different levels. [32]

There is no universal standard for abstraction models. and they should be tailored by an organization to fit specific needs. Gorscheks abstraction model included four levels described in table 2.3[32].

Abstraction	**Description**
Goal (Product level)	Most abstract level. Comparable to organizational strategies.
Feature	All the features that product support.
Function	Functional requirements. What the user can do.
Component	The lowest level of abstraction. How something should be solved.

Table 2.3 *Gorscheks requirements abstraction model*

All the abstractions should have parent-child relationships to link lower-level requirements to higher-level goals. Another abstraction model by Leffingwell uses abstractions described in the table 2.4[36]. The table is ordered by the abstraction level so that the highest abstraction is on the top line.

2.2.3 Prioritization

Prioritizing development activities is one of the characteristics of agile development methods (subsection 2.3.2). It is also a crucial question in the new product development[8] and the requirements management literature[29].

AL-Ta'ani and Razali defined a conceptual framework for prioritization in agile development[37]. They researched different factors that contribute to the process

Abstraction	Description
Investment theme	Strategic investment themes for enterprise. Each theme is a strategic business area where investments can be allocated.
Epic	Large-scale development initiatives. Epic realize value of investment themes. Highest level of abstraction to be used to coordinate development.
Feature	Functional requirements. What the use
Story	A description of action that system needs to be able to for the user.
Task	The lowest level of abstraction that describes a necessary activity to complete a story

Table 2.4 *Leffingwell requirements abstraction model*

and categorized them into three parts; environment, process, and product. The environment includes identifying stakeholders, project constraints, and the nature of requirements. The process part contains defining different processes, such as selecting a prioritization technique. The last part, the product, is where all these processes are applied to generate high-quality requirements.

Several techniques to prioritize requirements in the backlog can be found from the literature. On such technique is MoSCoW, which is acronym from categories it uses; must have, should have, could have and won't have[34]. The idea is to label requirements with these categories but it does not provide tools to prioritize inside these different categories.

2.3 Agile development

There is still no generally accepted definition what agility actually includes[38]. However, a major difference to most traditional models such as waterfall, is that agile development methods are adaptive as traditional methods are predictive[39]. This section introduces the foundations of agile development methods and some key characteristics.

2.3.1 Agile principles

Even though many development methods such as extreme programming and feature-driven development that are nowadays considered as a part of agile development[40] have been developed as early as 1996 and 1997[35], the foundation of agile development principles was defined 2001 when seventeen software developers signed and

published Agile Manifesto[41]. The manifesto is often cited in agile development-related literature, and the website https://agilemanifesto.org mentions hundreds of supporters who have signed up to those principles after original seventeen developers. According to the manifesto, the values for agile software development are following.

- Individual and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration
- Responding to change over following plan

The manifesto also defines twelve different principles that emphasize collaboration between stakeholders, short and quick development cycles, self-organizing teams, and focus to satisfying the customer[41].

2.3.2 Agile development methods

Some general characteristics[10] for agile development are

- Iterative and incremental development process
- Progress measured by completed features
- Requirements are re-assessed and re-prioritized on each iteration
- Involving customer to development
- Cross-functional teams
- Open and flexible design

These days agile development methods are widely adopted among software development and achieving good results[40]. It seems that commonly referred to agile development frameworks such as Large-Scale Scrum (LeSS) and Scaled Agile Framework (SaFe) are all designed for the IT industry[42]. However, some examples of applying agile development methods can be found from other industries such as finance[43][44].

2.4 Portfolio management in an agile environment

As agile principles highlight the importance of interactions and interactions over processes and portfolio management is much focused on describing strict processes, they can easily be seen in conflict with each other. This section introduces case studies and other available research that have been made about combining agile development methods to the portfolio management process.

2.4.1 Case studies

Rautiainen, von Schantz, and Vähäniitty introduced portfolio management practices to PAF, who was already using Scrum as a development method[45]. They provided only initial results, but they were able to reduce the number of ongoing projects dramatically from 214 to 30, improve planning practices, and continue to use Scrum. The biggest challenge they discovered that employees had difficulties in understanding the difference between value-adding work and planning of the project due to a conceptual mix of agile framework and traditional project management. Their portfolio management process combined elements from Stage-Gate model described in subsection 2.1.3, the Open Unified Process, and PMBoK[5].

Karlsrtöm and Runerson did a case study about combining agile development methods with traditional Stage-gate project management[14]. Their initial setup was different as all three cases involved a company that has been using Stage-Gate models as a project management process, and companies were trying to adopt agile development methods. They found that State-Gate context and agile methods are compatible with each other. Introducing development methods improved the development process in all studied cases though not without problems. The problems encountered were mostly related to communication, but they concluded that such problems are common when introducing a new method. The quality of deliverables increased, and teams were able to focus on their current work track better than before. However, this study is focused on project management level and does not address the issue of managing portfolio but still provides encouraging results about combining methods from agile development and traditional project management. Stage-Gate was also one of the models that were applied in the study made in PAF.

In both case studies done by Rautiainen et al. and Karlström et al., it was noticed that the problem of so-called requirements cramming was reduced despite different baselines for studies. Requirement cramming means that extra features are squeezed in if requirements change during a project without adjusting originally planned

schedule. This is interesting since the initial setup, and the goal was different for these studies but managed to resolve the same problem.

2.4.2 Agile portfolio management frameworks

Frameworks for managing portfolio while applying agile development methods can be found from Vähäniitty[46], Leffingwell[36], and Krebs[23]. However, there seems to be no evidence that any of these frameworks are used in practice[2].

Vähäniitty describes an approach that can be used to establish a portfolio management framework in four steps[46]:

1. Identify development activity types
2. Set targeted spending levels
3. Identify suitable ways to manage and control different types of development activities in terms of rhythm and synchronize control points
4. Identify control points to govern the portfolio

He suggests three different three types of control points for portfolio-level management. **Roadmap revisions** should be established to focus on product vision and other long term planning. **Portfolio reviews** that set the rhythm for a development organization and look at current development activities as a whole and are the primary method to link operations with business strategy. The third control points are **fire brigades** that act like event-triggered portfolio review and are held to make mid-increment decisions. He also highlights **Progressive refinement**, which means that as an abstraction model has been established, all work items can be split from the highest abstraction level to the lowest level development activities[10]. This makes it possible to manage development activities on different levels. For example, using Leffinwell's model, the portfolio management can focus on epic level activities, product management on features, and developers can manage task level activities. This helps to connect high level business goals with the development work.

In his book, Leffingwell suggests using investment themes and epics as a primary way to model the portfolio management process (see Table 2.4 for a complete abstraction model)[36]. Epics will have their own portfolio backlog, which is used to identify important development activities by prioritizing them according to business strategy. Epics are managed in four different states called funnel, backlog, analysis,

and implementation. The management of development activities is done on three levels. Epics are managed periodically by a portfolio management team. Feature level artifacts in the abstraction models are managed by the product manager and product level. Stories are managed by a product owner together with a development team.

2.4.3 Challenges

The new product portfolio management literature is generally plan-driven and assumes that development work can be carefully planned and then executed accordingly as a linear process[10][2]. This leaves little room for changing requirements or unexpected delays. The agile approach is to embrace the change and adjust accordingly to achieve the best possible outcomes and emphasizes iterative processes. Because of these conflicting approaches, the portfolio management literature is often not compatible with agile methodologies, since the continuous feedback loop required might be hard to achieve, especially with project management models using strict linear phases like the waterfall model, which do not allow efficient iteration process[47]. On the other hand, agile development literature is often focused on project level agility and won't provide solutions for larger-scale portfolio management[38]. This problem has been identified, especially in the software industry, and many studies about scaling agile development can be found[42][26][48].

Stettina and Hörz have done an empirical study about utilizing agile portfolio management, and they described three main challenges that they observed[2]. The first was alignment with existing processes like existing project management, software management, and business practices. The second was commitment, especially from senior management. The third challenge was resource allocation. The first one is in line with findings with PAF case study as they had a hard time understanding the conceptual mix of agile frameworks and traditional project management[45].

Another empirical study by Ahmad, Lwakatare, Kuvaja, Oivo, and Markkula identified three issues based on interviews about porfolio management made in companies applying agile development methods[49]:

1. *Difficulties to determine the long-term and short-term value of proposed initiatives.*
2. *Multiple offerings in the portfolio, as well as the increasing number of proposed initiatives, made it difficult to prioritize and balance them against available capacity.*

3. *Balancing the time factor was challenging for the companies, mainly because the portfolio was constantly exposed to change and evolution.*

These findings are also in line with other research reviewed. The first one is a relevant issue in the general portfolio management process. The second one was also noted by Stettina et al. Based on the literature reviewed, and the third one seems to be a challenge especially in agile environments.

2.5 Summary

The literature review was a combination of portfolio management in new product management, requirements management, and agile development literature. Based on the literature reviewed, it seems that some of the challenges found are encountered in the portfolio management process regardless if a company is applying agile development methods. Some key challenges that need to be resolved are:

1. Decision making process
2. How to select optimal projects to the portfolio
3. How to measure the portfolio performance
4. Estimating priorities, business value, and risk

If an organization is utilizing agile development methods, it is important to integrate portfolio management with development practices. The portfolio management process should be iterative and dynamic so that it will keep up with the changes happening on the project level and won't be in conflict with project management. This causes some challenges for the management as outputs for the projects are not as strictly defined as in traditional project management practices. In order to pursue organizational level agility, it is not enough that project management uses agile development methods; management processes on top of it must adapt as well. This requires commitment from the management as they need to change their working habits.

A concrete action towards agility in project management is to establish recurring portfolio reviews and other meetings to ensure a short feedback loop to synchronize development activities with current business requirements. Progressive refinement of requirements seems to be an important process to link abstract business requirements with actual development tasks. This requires an abstraction model that

describes requirements on different hierarchy levels and enables planning processes at multiple levels. The top-level being strategic decisions made by management and the lowest concrete development work done by agile teams.

3. THE DESIGN OF THE NEW DEVELOPMENT PORTFOLIO MANAGEMENT FRAMEWORK

This chapter introduces a development portfolio management framework designed for the case organization. The framework is designed based on the literature reviewed, and employees interviewed.

The terms used in the portfolio management framework are familiar from agile related literature and commonly used project management tools like Jira[50]. All the terms are defined in the context of this new portfolio management framework, so that they might be defined differently in the literature.

3.1 Guiding principles

There were a few important requirements and principles that the development framework needs to meet.

- Supporting agile development methods
- The framework should be simple, dynamic, and lean
- Used to organize small scale in house business development projects
- Possible to implement without changing organization structure

At the beginning of the project, some issues identified as described in the background section were lack of communication, prioritization, resourcing, and the heaviness of the existing official project management model. For these reasons the agile principles emphasized were

- **Collaboration** to improve communication by working together

- **Short development cycles** to help dynamically manage resources and prioritize development efforts
- **Self improving by default** so process can be developed further incrementally

The principle to keep model simple was set because the scope was to manage small scale in house projects, and the hypothesis was it would be less painful to implement while still achieving a targeted amount of business value.

3.2 Planning process

The first phase of designing a portfolio management framework was to gather information to gain an understanding of the current situation. Sources of information included employee interviews and existing internal documentation about previously used project management methods. Ongoing development activities were documented to identify activity types inside a portfolio.

Once an understanding of the baseline was established, the project team began the design process using literature and expert opinions. The most important reference models were Stage-Gate project management mode and agile portfolio management frameworks described in subsection 2.4.2.

The project team was constantly trying and applying new ways of working, such as using kanban, scrum like sprints and periods, to see how different approaches would work. There was also an ongoing in-house software development project that was used as a test group to test how events like retro and a demo worked and were perceived.

The third phase was to present new ideas to different audiences and gather feedback. After feedback was received, the ideas were re-evaluated and modified accordingly if needed.

3.3 General description

As the investment organization includes a relatively small number of employees, it was estimated that a quite simple portfolio management framework should be enough to achieve the goals set by management. Also, the organization had only a limited amount of experience in project management practices, so implementing a complex framework was estimated to be challenging. Assumption was that relatively

quick deployment, and training phase should be the best way to achieve the best business value.

For the reasons above, it was decided that a simple structure containing only one development portfolio containing all current development epics should be sufficient enough. The research review also suggested that the benefits of using programs as a way to organize development projects inside the development portfolio often do not realize the targeted benefits[12].

A subset from the abstraction model defined by Leffingwell is used to abstract development activities and allows progressive refinement. Development planning is done on many levels corresponding to abstraction, and main management responsibilities are also divided based on those levels. Stage-Gate model was also adopted to control the process from development idea to production as it was applied in many cases in the literature. Development activity abstractions are managed in five different stages, which are used to control the workflow of work items from idea to production. Stages are described in section 3.5.

The model has three different levels that correspond to selected abstractions. Those are

1. Portfolio level to manage epics
2. Product management level to manage features
3. Team level to manage development work in practice

Each level has its own planning and managing responsibilities, which are defined by roles in section 3.4. It was chosen not to use a separate PMO, but instead, PMO responsibilities are handled by the portfolio management. Mandatory recurring meetings, such as weekly portfolio reviews, were defined to ensure short feedback cycles and iterative management process. This was suggested by the framework described by Vähäniitty[10]. All defined recurring meetings are described in section 3.7.

3.4 Roles

This section describes what roles are included in the portfolio management framework and what are their main responsibilities.

3.4.1 Portfolio management

A group formed by management that is responsible for prioritizing epics and managing the amount of work currently in progress on portfolio level. Portfolio management has control over states of different epics, and each time an epic is transferred to the next phase, it requires active decisions made by the portfolio management. They also have the responsibility to allocate development resources to reflect company strategy set by top management and to produce the maximum amount of business value. Compared to the traditional stage-gate project management model[16], portfolio management can be seen as a gatekeeper for control points in the portfolio management process.

3.4.2 Product management

Each epic is assigned an owner at least before it can be transferred into in-progress state for implementation. Epic owners are responsible for the product management level of the framework. In the context of the case organization, the term product management is confusing as internal development initiatives do not focus on creating a new product. Content management might be a better term to describe this management level, but as product management is generally used in literature this term was chosen.

By default, an owner has a business ownership and a responsibility to organize development activities and resources needed to complete the assigned epic. Responsibilities also include that work that has been done actually satisfies the requirement and make sure that all activities are contributing towards completing the parent epic.

The role is crucial, so the management needs to make sure that the selected person has been allocated enough time to fulfill his duties. The decision to delegate business and project management responsibility to a single person was made because most of the epics are such a small scale that the work required to manage is limited. If an epic is a more large scale these responsibilities can be divided inside the epic team.

3.4.3 Development teams

Group of individual employees identified and needed for their expertise to complete an epic. Teams are not fixed but instead are identified case by case. An epic can not

be transferred to the in-progress state before the development team for it has been identified. The epic owner assigned by portfolio management is responsible for identifying required team members based on needed expertise and available resources.

A person could be part of many different development teams simultaneously, but it is not recommended to participate in more than two epics at the same time. Teams should be formed in such a way that it includes expertise from all different functions in investment process. In other words, the aim was to form cross-functional teams, which is also emphasized in the literature related to agile development methods. Cross-functionality was seen as a way to increase communication between different functions and also as a way to increase organizational learning.

3.5 Work item stages and workflow

Each work item has a state which corresponds to a stage that it is currently at. The state of the item is managed using a kanban board. The portfolio management framework introduces five different states work item can have. Stages are described in general on table 3.1. It was decided to use stages that would be self-explanatory as possible to employees. It was assumed this would make it easier to introduce those. For example, it was chosen to use stage an idea instead of a funnel that is used in SaFe-framework.

State	Description
Idea	Idea state is the initial state for most issues though it is usually most relevant for epics. Anyone can submit idea to Jira and is free to argue for it on epic owner weekly meeting where all new epic ideas will be dealt with. Only submitting issue to Jira is required for this state.
Analysis	Portfolio management approves idea for further analysis if they estimate it could achieve business value. In this stage idea is refined with more detail and details required to start the is gathered. Documentation will be made to confluence. Documentation is only required for epics.
Backlog	All the issues ready to be worked on.
In progress	All the issues that are currently being worked on.
Done	Issues which have been completed.
Rejected	Work items that has been rejected for some reason.

Table 3.1 *Different issue states*

Figure 3.1 describes the generic workflow for work items. In practice, workflow depends on the work item type. Following section explains in detail how the workflow is applied for different work item types.

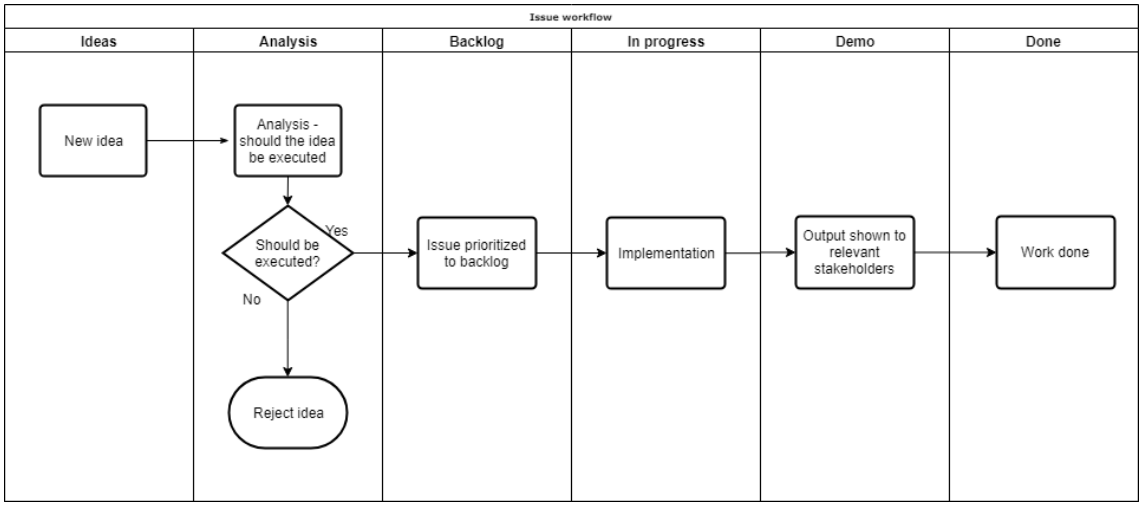

Figure 3.1 *Generic issue workflow*

Epic kanban board acts like a light version of a stage-gate model where portfolio management is controlling the gates. Before an epic can be moved from one state to another, there must be an active decision made by the portfolio management. In practice, this happens in a development portfolio management weekly meetings described in section 3.7.

3.6 Abstraction model for work items

Abstraction model describes what work item types are used and how they relate to each other. A model defined by Leffingwell[36] was chosen as a baseline with a few modifications. It provides multiple levels of abstractions and uses the same terminology that is also used in many software tools like Jira, agile literature, and agile development frameworks. The hypothesis is that by using commonly used terms onboarding, new employees to the portfolio management model will be easier if they have previous experience with agile development frameworks. However, it was chosen to use only a subset of abstractions, for now, to avoid confusion since abstraction levels were not familiar to most of the current employees. Task and investment themes were dropped for now though the latter one was considered to be implemented. It is possible to implement investment theme and task as the organization matures and if there will be a need to scale abstraction levels up.

These all abstractions are forming the portfolio, but on the portfolio management level, the main focus is on the highest level of abstraction, which are epics.

3.6.1 Epic

An epic is the largest unit of work used in the model and the highest abstraction used in development work. The most major difference to traditional projects defined by Project management institute is that projects have a definite beginning and end[5], but by default, epics will continue until portfolio management approves that it is done. The assumption is that requirements can and will change during execution, so there is no point in setting strict deadlines as the scope and amount of work required to complete the epic might also change. This allows iterative workflow compatible with agile development methods.

Even though an epic has no fixed schedule by default, there might be strict deadlines if needed, for example, because of a need to meet requirements set by regulation on time. Epic usually takes at least one period (see section 3.8) to complete, and the maximum is not defined.

After the idea for epic has been recognized, the first thing to do is to define the content by creating a **epic template**. It is a fixed field form that requires answers to the following questions:

- Description of current situation regarding development target
- What development work is proposed to be done
- How does it bring business value
- Success criteria that should be met before epic can be considered done
- What expertise is required to execute it

Once the epic template has been made, the epic is ready to be submitted to the epic kanban board to the idea state. These steps can be done by any employee. The person who invented the idea will be asked to present it in the next weekly portfolio management meeting. After the presentation, the portfolio management group will make a decision if epic should be transferred to the next state, which is an analysis following the work item workflow described in 3.5. Idea can also be abandoned at this point or left to idea state to be decided later.

On analysis, the state epic is evaluated more in detail. This includes further analysis of business value, resource requirements, and if capabilities to complete the epic are available. Often the scope and the content of the epic template are redefined during

the process. After an analysis has been completed, epic is reintroduced to portfolio management. They will make a decision if epic should be moved to the backlog, epic should be rejected, or if it still needs further analysis.

Once an epic reaches backlog state, it is prioritized by the portfolio management. Priority will be evaluated weekly and can be changed. Portfolio management will eventually transfer the epic into the in-progress state, depending on priority and available resources. Before an epic can be moved to in progress, it should have assigned an epic owner and identified the development team.

Once in progress, actual development work will begin. The epic will be followed up regularly by portfolio management until it is completed and will be moved to a stage done. Usually, the results have to be demonstrated to the portfolio management before the decision can be made. Before transferring to the stage done, the management must make sure that there are no unfinished features related to it.

3.6.2 Feature

A feature is a unit of work split from an epic. A feature should always be a logical high-level requirement that advances the epic it belongs to towards completion in some way. It should take no more than one period (three months) to complete by estimation. It is recommended to specify success criteria and achieved benefits before starting to work on it.

The feature can follow the same generic workflow as epic, but decisions to move features to different states are made by the epic owner and the development team instead of the portfolio management. It is also allowed to simplify workflow by only using stages backlog in progress and done if seen fit.

In a backlog, features should be prioritized and moved into in-progress stage by pulling the next feature, which has the highest priority from the prioritized backlog queue when resources are available. Before transferring features to the done stage, it should be reviewed by the development team and the responsible epic owner. If a feature has stories attached to it, all of those should be done and reviewed before a feature could be considered as done.

3.6.3 Story

Story is the smallest unit of work divided and a part of a feature. It should be a small concrete step towards completing the feature it belongs to. The amount

of work should be possible to complete by one person during a time period of two weeks.

Managing stories is left to be decided by the development team and the assigned epic owner. Teams are free to use other tools than Jira like physical boards as the management won't generally track development on this level of accuracy. If a team is using Jira, it should use the same stages or subset of those used in the generic model. If they choose to use other tools, they are free to use any stage convention, and they see fit.

3.6.4 Relations between work items

The figure 3.2 illustrates how epics are composed of features, and features are composed of stories. It also shows how the primary responsibilities for different parts of the framework are organized.

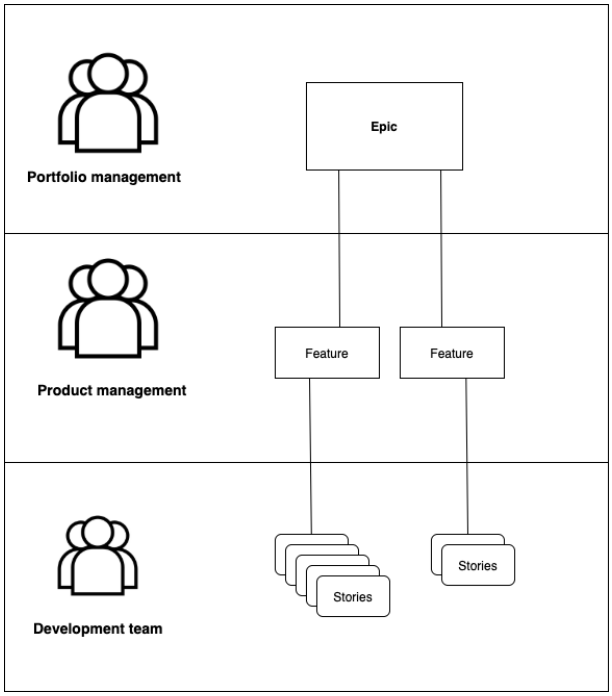

Figure 3.2 *Relations between work item abstractions and roles*

The epic owner has overall responsibility for the progression of the epic. This means that he or she has at least the business responsibility by default but also the responsibility to organize work and push the epic forward. It is still possible to assign a separate project manager to help coordinate working, especially if epic is quite large in the amount of work and requires a large development team. In this case epic owner only has the business owner's responsibilities.

Every epic owner for an epic that is currently in progress form the second level, which is referred to as the product management in the diagram. The main work item type of epic owner is working with is a feature. Features are planned together with the development team, and the epic owner has the responsibility to follow up on how they are progressing.

The last level is the so-called team level. A team member is mostly working with different stories, which are work items that are part of a feature. The story is the smallest work item used in the model, and one story should require only a limited amount of work that can be assigned to a single team member. A story should take no more than two weeks to complete.

3.7 Recurring meetings

Recurring meetings are a crucial part of planning development work in all levels, communication, and enabling iterative workflow. All fixed recurring meetings are presented in the table 3.2.

Event	**Participants**	**Frequency**
Period planning	All employees involved with current development work	Once in three months
Period demo	All employees involved with current development work	Once in three months
Period retro	All employees involved with current development work	Once in three months
Portfolio management weekly	The portfolio management group	Weekly
Epic owner status	All active epic owners	Bi-weekly

Table 3.2 *Recurring meetings*

3.7.1 General meetings

The most important meeting for the development framework is a **period planning event**, where work is identified and planned for the next period. **A period** is defined as a timeframe of three months. Period **retro** and **demo** events are also facilitated just before planning. These are mandatory for all teams and other employees working somehow related to current development activities. The idea of retro events is to

gather feedback, noticed issues, and think of actions on how to improve our development processes further. Systematical collection of feedback enables the iterative development of the portfolio management process itself.

3.7.2 Portfolio management meetings

The portfolio management group has status meetings weekly. The agenda is to go through the statuses of epics, re-prioritize if needed and make decisions about the future roadmap for epics in the backlog. New ideas emerged are also presented in this meeting to be decided on how to continue with them.

3.7.3 Product management meetings

Epic owners have regular status update meetings with each other in bi-weekly **epic owner status meetings**, and also, the portfolio management group can participate. The agenda is to share information about current progress, discuss resourcing, and if any issues have occurred. The meeting is producing information to the portfolio management so that they can adjust resourcing according to current priorities if needed and make other decisions about ongoing development activities.

3.7.4 Team level meetings

Because teams in our model can be quite different from each other, strict guidelines were not introduced for teamwork. The framework recommends that teams will at least have regular status meetings and kickoffs if epic is starting, but otherwise, teams can use scrum or other practices as they see fit.

3.8 Planning development

There are two primary ways to plan development on the portfolio level. The primary planning event is period planning, where a large group of Elo's professionals gathers together to workshop around different epics that have been previously recognized. Planning generates information for the portfolio management group, which makes final decisions about the development work in a weekly development portfolio management meeting.

The portfolio management meeting is held once a week, and current epics are continuously evaluated. So a decision to start, kill, or put on hold can be made weekly basis

by the management. Employees are encouraged to present new development ideas, and they are presented to the portfolio management in the next weekly meeting.

Using progressive definement as described by Vähäniitty's framework (see subsection 2.4.2), an epic is split to features and eventually to stories. This and managing schedules for an individual epic is managed by the epic owner with the team assigned to work on it. By default, there are no fixed schedules set by the portfolio management, but these can be set if required and set at least an approximate deadline is encouraged. In case there is no fixed schedule, the responsibility to manage an epic timeline is on the epic owner. If the portfolio manager wants to hurry up the progress for a single epic, it should do it by prioritizing the epic higher and by allocating more resources to work on it. The fundamental assumption behind this idea is that employees are working fast as they can by default, so actively allocating more resources is the best way to accelerate progress instead of applying more pressure to the current development team.

Even though usually there are no fixed schedules, rough estimates are made during the planning events on feature level and if otherwise required by the management. The year is divided into four periods, and an epic manager with his team tries to schedule all single features in those periods. Sometimes features overlap between periods, but the time period required to complete it with available should be less than three months. If a feature includes such a large amount of work that it can not be completed within that timeframe, it should be divided into two or more smaller features.

On a team level, work included in the features is planned on the story level. Development teams can use a scrum-like sprint model or other agile development methods if they choose to do so. This is recommended but not required as teams can decide by themselves how they want to manage tasks inside features.

3.8.1 Prioritization

Prioritization in the portfolio is done by the portfolio management group. All epics are prioritized relative to each together based on a mutual discussion. Using a formal framework to prioritize epics systematically was considered, but it was decided not to use one for now. As an input portfolio, management uses opinions from employees, which they are free to present during the portfolio management weekly meeting. A period planning event is also an important event where information for prioritization can be gathered. Priority for epics is set using a rough scale described in the following table 3.3.

Priority	Description
Critical	Serious impacts to business if not completed. For example failure to be compliant to changing regulation in time. Often has strict deadlines and considered mandatory for continuing business.
Major	Potential to produce significant business value or major improvement to operational efficiency. For example automating some part of process to improve quality or to release time from employees.
Minor	Only minor business or operational value. Usually no deadline or could be delayed further without major impact to business. For example implementing new analysis tool.

Table 3.3

3.8.2 Resourcing

As most of the employees have some daily work not related to actual development, the first question is how much resources are allocated to development work and how much for daily work. The framework does not address this issue, and this is left to be decided by employees themselves with their supervisors.

In the scope of the portfolio management framework, employees have a big responsibility to allocate their time, so that it produces the best value for the organization. Portfolio management can change priorities for epics, and this should obviously reflect resourcing. The big picture of resource allocation inside the development framework is created during period planning events, but the situation can change during periods. Estimates also have a fair amount of inaccuracy, so the big picture is mostly directional. Realization is compared to estimates in period retro events, so presumably, the accuracy of the estimates will improve incrementally.

3.9 Tools and systems used

All documentation related to development work, including epic templates, are stored in Atlassian Confluence[51]. Work items and their statuses are maintained in Atlassian Jira [50]. Some planning is done using physical walls and post-its. In the beginning, physical walls were much used, but as the number of issues grew larger, maintaining walls became too cumbersome. Reporting was done using Jira, and some custom Tableau[52] reports.

4. DEPLOYMENT

This study followed the adoption and deployment process in the case organization for four months. Some results and qualitative evaluation is described in this chapter.

4.1 Deployment plan

Deployment followed loosely steps suggested by Vähäniitty[10]. Steps were designed for setting up portfolio management for small software organizations, but they were seen as general enough to be used in the current context as well. Quoted steps are the following:

1. *naming a group of people to be responsible for portfolio-level decision-making*
2. *building a publicly visible list of all ongoing activities that require time from development,including the information on who are assigned to which activity*
3. *synchronizing the portfolio*
4. *meeting regularly at portfolio sync-points (for example, on a bi-weekly basis) to keep the list of ongoing activities up-to-date, perform short-term prioritization (force-ranking the ongoing activities) and setting the default resource allocation until the next meeting*
5. *agreeing on how decisions affecting more than one ongoing activity are made in urgent, 'emergency' type situations*
6. *Identifying the different types of development activities*
7. *setting target spending levels per development activity type that reflect the organization's strategy, and possibly tracking the actual spending*
8. *curbing excessive multi-tasking by explicitly setting limits to the number of concurrent activities a person can be involved in*

9. *tracking the developers' (or development teams') workload*
10. *ensuring that incentive systems do not steer people towards local optimization*

It was decided not to put development activities into different categories as in step six to keep the framework as simple as possible, and as budget and spending was out of the scope the step seven was not included. Setting a specific limit for current activities for a single person, as stated in step eight, was also ignored because the nature and the scope of development activities vary so much that it was seen as a not meaningful way to track workload.

While the development project for creating a new portfolio management framework was still in design phase, the project team hosted multiple roadshow presentations to different parts of the investment organization. The focus was to introduce general concepts like information model and roles used in the model.

Initially, it was considered that two different pilot groups would be launched, and the framework would be adjusted based on the feedback. This plan was abandoned because it was estimated that most of the business value would be achieved by gaining control of the overall development portfolio because it enables the management to focus efforts to meet the company's strategic goals.

The plan that was utilized was to kick-off the new framework at the beginning of the next year and train new practices by coaching employees as epic teams started working. Basic concepts were repeatedly presented on different occasions to put new ways of working into practice. Hypothesis was that we could familiarize organization with basic concepts at first by emphasizing just things like defining epics the right way and splitting work into features we could introduce more difficult concepts after the organization had learned to manage with simple ones.

It was noted that software systems Jira and Confluence were also new to most of the employees who could cause an additional level of difficulty in adopting a new process. After evaluating tools, it was decided that Jira and confluence should be configured in such a way that it would be easy as possible to start using those. Some written documentation was also produced to support usage.

4.2 Introducing the framework to the organization

This section describes how the framework was introduced to the organization. This includes preparations before the actual launch and how the first recurring meetings were established.

4.2.1 Preparations before introduction

Before the new portfolio management model could be implemented in practice, some preparations were needed. The first thing was to actually map ongoing and currently planned development projects so they could be organized using the model. This is required by steps two and three in the implementation plan. Information was gathered by interviewing management, trading desks, and other employees. After initial results, a workshop event was organized to define the content of those development epics more precisely. Documenting epic templates, which was defined as a first step to start the epic workflow, was done during the workshop. As a result, we managed to get 14 different epic templates filled with information defining the content.

The initial plan was to prioritize some subset of epics from those which had been created during the first workshop and introduce them at the first-period planning event. Eventually, we had to change our plans because the management saw that there were still some important tasks unidentified and not formed as epics, so they work towards identifying all relevant epics continued. Eventually, we gathered the most relevant epics, and the management prioritized the 12 most important ones to be introduced in the first-period planning event.

Identified epic owners had a role in the first planning event, so they had to be prepared before the event. This ended up being one of the biggest challenges as the decision about selected epics was made only one week before actual planning, which left very little time to arrange all preparation meetings. Time of the year was also challenging, as many employees were still on a Christmas holiday in the beginning of January.

4.2.2 Introducing the framework

The first period planning event was held 8th of January 2019, and it was also an official launch for the new development portfolio management process. 12 different epic were preselected for the event, and they were planned on four different working areas. Workgroups created features from the epics and planned for tentative resourcing. The teams were self-organizing as epic owners and employees recognized by themselves which epics might require their expertise.

After all the epics where planned, those were presented on a large board presenting timeline. Based on the board, a public discussion was held to evaluate how much of the work identified should actually be started simultaneously and which epics should be left to the backlog. It was noted during the planning that the amount of

work planned for the first period seemed unrealistically high. The management was able to use this information to help in making final decisions regarding priorities and order of implementation. Portfolio management made decisions the next day, and four of the twelve epics were put to backlog on portfolio management weekly meeting.

Overall the first planning event was a great success and received excellent feedback, which was gathered at the end of the day. The gathering method was simply to require everyone to write their short feedback on a post-it note, which was put on a wall for everyone to see. Based on the feedback, it was found that the most important part of the event was sharing information between different working units.

4.2.3 Kick-offs for first epics

After the first planning, every assigned epic owner that had an epic in progress held a kick-off event for the team identified during the planning. The agenda of the meeting was to go through the results from the planning event and agree on how the team should continue working on this epic. This included agreement of regular meeting schedules and division of labor. Most of the teams got started off quickly and started actual work as soon as possible. In some cases, the kick-off event was delayed for a few weeks because epic was not specific enough to start working right away, so it was decided to do some preliminary analysis work before the actual kick-off. Eventually, all teams started working successfully.

4.2.4 Establishing recurring meetings

After the framework had been introduces, recurring meetings described in were arranged regularly 3.2. Feedback about the framework was gathered during the portfolio management weekly meeting and bi-weekly epic owner status meetings.

The second so-called period planning event was held in April three months after the first one. Preconditions for the second planning were different as during the first planning, and every epic was treated as a new one even though in many cases, some amount of work was already done. The planning day had three separate phases.

1. Demo - what had been achieved previously
2. Retro - collecting feedback and actions to improve working
3. Planning - actual planning of development work for next period

This is the structure that is established for the next planning events as well.

On the demo phase, epic owners from each epic that was in progress during the last period introduce their results. Presentations are short, and it is instructed not to put too much effort into those. A demo is not a formal reporting event and should focus on the actual results.

The retro phase is a simple feedback gathering event. The basic pattern is to hand out a few sticky notes where participants write feedback on two simple categories. What has gone well and what things could be improved. The focus is to gather information about the portfolio management process itself to understand how the new framework is affecting development work. After having been written, those are gathered and put on a wall for everyone to see. All similar feedback is grouped together. Based on those groups, there is a common feedback discussion. Some most common feedback received during the second period planning was:

- Need to communicate better about decisions made by portfolio management
- Employees were not yet comfortable with new software tools used
- Need more focus towards planning future development roadmap
- Visibility to ongoing development had increased

The planning part follows the same structure as in the first period planning. It is recommended that epic owners establish similar events for the development teams for individual epics to integrate the project management process with the portfolio management process.

5. RESULTS AND FINDINGS

The goal of this study was to find out from the literature how to implement project portfolio management processes in a way that supports agile development methods and find out if agile principles can be applied to the process as well. Based on the findings, a complete portfolio management framework was designed and implemented in the Elo Mutual Pension Insurance Company's investment organization. This chapter describes conclusions regarding original research questions, evaluates if the framework was able to resolve identified issues.

5.1 Research questions

This section reviews original research questions and answers based on the literature review.

RQ1: *What adjustments to portfolio management should be made to make it compatible with agile development methods?*

Based on the literature reviewed, the most significant adjustment needed is to support iterative and incremental processes used in agile development. Portfolio management can not rely on preplanned strict linear processes. This makes it harder for the management to plan schedules, so commitment from the management is essential if organizations are going to apply agile development methods.

RQ2: *What agile principles could be applied in the context of Elo Mutual Pension Insurance Company's investment organization to increase agility in the portfolio management process?*

As requirements and schedules are changing, the portfolio management process should establish periodic portfolio management reviews to keep up with the changing requirements and adjust resource allocations as priorities change. The process allows iterative and incremental development, which is one of the fundamental values in agile development.

Defined recurring meetings and progressive refinement increased collaboration and

shortened feedback loops. These are also values highlighted by the agile literature.

RQ3: *What are the first actions that should be taken in an effort to introduce agile portfolio management?* Identifying all development activities is a crucial first step, which helps to design proper abstractions to model development activities. The company must identify management processes to these activities and set up control points. The management process should be periodic with short cycles to ensure that it is compatible with the iterative nature of agile development methods. Integrating all development related processes together is crucial to achieve the benefits of agility.

5.2 Qualitative evaluation of the new portfolio management model

According to general feedback from management and employees, the portfolio management framework was overall considered as a beneficial improvement to the organization. Feedback was gathered during planning events and also in retro meetings with development teams and management. As all these results are based on feedback and observations within relatively short period of time, they should be considered as preliminary.

5.2.1 Achieved improvements

On a portfolio management level, the framework was able to increase understanding of current development activities and to control these activities better than before. Even though some challenges remained, the framework helps to identify these and design further actions to improve the process.

The framework increased visibility to ongoing development activities across the organization. This enables more efficient collaboration with employees and identifying needed resources for different development projects as employees are aware of what is going on. Even though prioritization remained an issue, at least employees have a better understanding of what is currently ongoing and where resources are allocated.

Establishing the abstraction model for development activities enables progressive refinement, which enforces the link between business strategy and development work. The framework is compatible with agile development methods as the portfolio management is iterative and ready to respond to changing requirements inside individual development activities.

5.2.2 Steps towards agility

The framework introduced several recurring meeting practices to be more responsive to changing environments and requirements. This enables iterative development and quick response to changing situations, which is part of the agile mentality. The portfolio management process itself is designed to be self-evolving as feedback is regularly gathered, and adjustments to the process are made.

The planning process of the portfolio is focused on identifying and prioritizing the most important development activities currently and adjusting resourcing accordingly. Based on the literature review, this approach is more compatible with agile development methods than focusing on linear processes and preplanning. However, at this point, it is too early to draw a conclusion if the framework is compatible in practice as well.

The portfolio management process is tightly linked with project management processes[27], so it is a fair question to ask if the portfolio management process can be considered agile even though some of the projects inside the portfolio do not apply agile project management methods. In some cases, it was observed the agility achieved in the portfolio management process could cause issues in individual projects if they are not adjusting well to the changing environment. If the portfolio management decides to re-allocate resources during a project relying on strict preplanning, the project manager (or the epic owner) might have problems with adjusting, especially when a project loses resources. So it is a question of debate if the designed framework can be considered agile, but observations highlight the importance of integrating process management processes with portfolio management to be able to achieve organization level agility.

5.2.3 Challenges

Some teams began to achieve results quite soon after the first planning event and worked intensively on their epics, but on some teams, epic owners were struggling to commit their teams. Many members of teams could only allocate a small portion of their time to development work, which was probably the main reason for lack of commitment.

Monitoring performance is hard as there are no standard metrics used to follow up team velocity. Though this was a conscious decision to minimize unnecessary reporting so this is an almost unavoidable trade-off. As everything was new to teams, not everyone split their work items to story level abstractions, but as the

organization learns the idea of progressive refinement, it can be assumed this will improve over time. When the habit of making stories evolve, measuring progress on story level might improve performance measurement in the future.

The portfolio management process should be able to allocate development work such a way that the portfolio produces maximum business value using scarce resources. An important part of this is prioritizing development activities. The framework does not provide enough tools to do this as well as possible at the moment. One factor is that as the nature of epics has large variability, they are often hard to prioritize relative to each other.

The framework was designed to be as light as possible, so we did not use dedicated project managers. But in practice, many of the traditional project manager responsibilities landed on epic owners, and as many of them had only little project management experience, and limited time, it became a heavy responsibility to organize teams. Though this was not considered as a major problem. It is possible to name dedicated project managers in a future for larger epics, and the actual epic owner could focus on product owner responsibilities.

During the implementation and training, it was decided to teach "on the fly" as development work progressed without formal hands-on training sessions. At the start, development teams were supported by coaches who were part of the team responsible for creating the new portfolio management model. In retrospective, this approach was not sufficient as putting to practice was not as easy as it was expected. Implementation included introducing software tools such as Jira that were new to many employees, and based on the collected feedback, they did not receive enough support for using these.

5.3 Conclusions

Based on the initial results, the case organization was able to benefit from a new framework designed to manage the development portfolio. Traditional portfolio management literature, project management, and agile portfolio management literature were applied to design the framework. It seems that many of the initially identified issues were resolved.

However, it is hard to distinguish what improvements were achieved because of applying agile portfolio management practices. It can be assumed that many of the issues could have been resolved using traditional portfolio management introduced by the new product development literature.

Assumed compatibility with agile development methods was achieved by making design decisions based on the literature reviewed regarding the topic. So currently, there is no empirical evidence about the compatibility in practice.

5.4 Further ideas for the portfolio management framework

A number of different ideas were left on a table because it was decided we should start with simple as possible initial framework which can be expanded as employees learn new ways of working.

One of the downsides for the portfolio management framework is that it does not track resource utilization accurately. So, later on, it should be considered if employees would start to log time on different Jira tasks, which would increase visibility to current workloads and make it possible to track issues more detailed level. A major downside for reporting time to Jira is that it can feel like unnecessary bureaucracy, and many of the employees responded negatively to the idea.

During the initial planning of the framework, there were considerations if we should add an additional abstraction level to epics called investment theme as used in Leffingwell's abstraction model. The idea resembles a program from traditional portfolio management, but it was abandoned as unnecessary. However, it might be beneficial to implement it in the future to help management to track down resources used in different areas of business development. An example of a theme could be operational efficiency in trading or analysis of asset allocation. This could help top management to put emphasis on specific strategic goals instead of focusing on tightly scoped development epics. Abstraction level could also be expanded on a low end to include task work item type if there is a need to organize development activities with more detail.

It should be considered that epics are categorized into different development activity types. This would help to prioritize them as different types could be prioritized relative to each other instead of comparing all epics inside portfolio relative to each other. Also, a systematic way to evaluate priority might help to make this process more straight forward.

6. SUMMARY

This thesis researched the current literature about portfolio management in new product development literature, agile portfolio management, and some parts of requirement management. A new portfolio management framework was designed based on the research reviewed for the case company. The goal was to design a framework that supports the use of agile development methods and applies what can be learned from the agile literature. The context of the case company was internal development projects, including varying types of development activities, such as software development and business development work.

The designed framework was deployed, and it was followed up for four months to get some initial results about the usage. Results were evaluated using qualitative methods such as gathering feedback from employees.

The framework was able to improve the portfolio management process, which is great for the original project. However, empirical evidence about using the designed portfolio management framework with agile development methods in practice does not exist at the moment, so it can not be evaluated. The portfolio management process has some similar characteristics as agile development methods, such as iterative and ability to react to changing requirements, but it is a question of debate if the process can be considered agile as a whole.

BIBLIOGRAPHY

[1] G. Pohle and M. Chapman. "IBM's global CEO report 2006: business model innovation matters". In: *Strategy & Leadership* 34.5 (2006), pp. 34–40.

[2] C. J. Stettina and J. Hörz. "Agile portfolio management: An empirical perspective on the practice in use". In: *International Journal of Project Management* 33.1 (2015), pp. 140–152.

[3] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee. "A design science research methodology for information systems research". In: *Journal of management information systems* 24.3 (2007), pp. 45–77.

[4] S. Rajegopal, P. McGuin, and J. Waller. *Project portfolio management: Leading the corporate vision*. Springer, 2007.

[5] P. management institute. "Project management body of knowledge (pmbok® guide)". In: *Project Management Institute*. 2001.

[6] R. G. Cooper, S. J. Edgett, and E. J. Kleinschmidt. "Portfolio Management for New Product Development". In: (2006).

[7] I. Van De Weerd, S. Brinkkemper, R. Nieuwenhuis, J. Versendaal, and L. Bijlsma. "Towards a reference framework for software product management". In: *14th IEEE International Requirements Engineering Conference (RE'06)*. IEEE. 2006, pp. 319–322.

[8] K. B. Kahn, G. Castellion, and A. Griffin. *The PDMA handbook of new product development*. Wiley Hoboken, NJ, 2005.

[9] R. G. Cooper, S. J. Edgett, and E. J. Kleinschmidt. "Portfolio management in new product development: Lessons from the leaders—II". In: *Research-Technology Management* 40.6 (1997), pp. 43–52.

[10] J. Vähäniitty. "Towards agile product and portfolio management". PhD thesis. Aalto University, 2012.

[11] G. B. O. of Government Commerce. *Managing successful programmes*. The Stationery Office, 2003.

[12] M. Lycett, A. Rassau, and J. Danson. "Programme management: a critical review". In: *International Journal of Project Management* 22.4 (2004), pp. 289–299.

[13] R. Cooper. *Winning at New Products*. Addisson-Wesley, 1986. ISBN: 0771551010.

[14] D. Karlstrom and P. Runeson. "Combining agile methods with stage-gate project management". In: *IEEE software* 22.3 (2005), pp. 43–49.

[15] L. Mulder. "The importance of a common project management method in the corporate environment". In: *R&D Management* 27.3 (1997), pp. 189–196.

[16] S.-g. international. *Stage-gate model*. 2019. URL: `https://www.stage-gate.com/stage-gate-model/` (visited on 03/02/2019).

[17] R. G. Cooper and S. J. Edgett. "Stage-Gate® and the critical success factors for new product development". In: *BP Trends, July* (2006).

[18] G. I. Kendall and S. C. Rollins. *Advanced project portfolio management and the PMO: multiplying ROI at warp speed*. J. Ross Publishing, 2003.

[19] P. Hodgkins and L. Hohmann. "Agile program management: Lessons learned from the verisign managed security services team". In: *Agile 2007 (AGILE 2007)*. IEEE. 2007, pp. 194–199.

[20] C. Carlsson, R. Fullér, M. Heikkilä, and P. Majlender. "A fuzzy approach to R&D project portfolio selection". In: *International Journal of Approximate Reasoning* 44.2 (2007), pp. 93–105.

[21] L. Kester, E. J. Hultink, and K. Lauche. "Portfolio decision-making genres: A case study". In: *Journal of engineering and technology management* 26.4 (2009), pp. 327–341.

[22] R. G. Cooper, S. J. Edgett, and E. J. Kleinschmidt. "New product portfolio management: practices and performance". In: *Journal of Product Innovation Management: AN INTERNATIONAL PUBLICATION OF THE PRODUCT DEVELOPMENT & MANAGEMENT ASSOCIATION* 16.4 (1999), pp. 333–351.

[23] J. Krebs. *Agile portfolio management*. Microsoft Press, 2008.

[24] K. Jensen and P. Warren. "The use of options theory to value research in the service sector". In: *R&D Management* 31.2 (2001), pp. 173–180.

[25] N. M. Arratia-Martinez, R. Caballero-Fernandez, I. Litvinchev, and F. Lopez-Irarragorri. "Research and development project portfolio selection under uncertainty". In: *Journal of Ambient Intelligence and Humanized Computing* 9.3 (2018), pp. 857–866.

[26] C. Larman. *Practices for scaling lean & Agile development: large, multisite, and offshore product development with large-scale scrum*. Pearson Education India, 2010.

[27] M. Martinsuo and P. Lehtonen. "Role of single-project management in achieving portfolio management efficiency". In: *International journal of project management* 25.1 (2007), pp. 56–65.

[28] M. Martinsuo. "Project portfolio management in practice and in context". In: *International Journal of Project Management* 31.6 (2013), pp. 794–803.

[29] P. management institute. *Requirements Management: A Practice Guide*. Project Management Institute, Inc, 2016.

[30] O. Lopez. "Requirements management". In: *Journal of Validation Technology* 17.2 (2011), p. 78.

[31] J. O. Grady. *System requirements analysis*. Elsevier, 2010.

[32] T. Gorschek and C. Wohlin. "Requirements abstraction model". In: *Requirements Engineering* 11.1 (2006), pp. 79–101.

[33] T. Sedano, P. Ralph, and C. Péraire. "The product backlog". In: *Proceedings of the 41st International Conference on Software Engineering*. IEEE Press. 2019, pp. 200–211.

[34] C. G. Cobb. *The project manager's guide to mastering Agile: Principles and practices for an adaptive approach*. John Wiley & Sons, 2015.

[35] S. Ashmore and K. Runyan. *Introduction to agile methods*. Addison-Wesley Professional, 2014.

[36] D. Leffingwell. *Agile software requirements: lean requirements practices for teams, programs, and the enterprise*. Addison-Wesley Professional, 2010.

[37] R. H. AL-Ta'ani and R. Razali. "Prioritizing requirements in agile development: a conceptual framework". In: *Procedia Technology* 11 (2013), pp. 733–739.

[38] P. Kettunen and M. Laanti. "Combining agile software projects and large-scale organizational agility". In: *Software Process: Improvement and Practice* 13.2 (2008), pp. 183–193.

[39] M. Stoica, M. Mircea, and B. Ghilic-Micu. "Software Development: Agile vs. Traditional." In: *Informatica Economica* 17.4 (2013).

[40] T. Dybå and T. Dingsøyr. "Empirical studies of agile software development: A systematic review". In: *Information and software technology* 50.9-10 (2008), pp. 833–859.

[41] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, et al. "Manifesto for agile software development". In: (2001).

[42] C. Ebert and M. Paasivaara. "Scaling agile". In: *IEEE Software* 34.6 (2017), pp. 98–103.

[43] G. M. Roche et al. “Agile portfolio management at NYSE”. In: *2012 Agile Conference*. IEEE. 2012, pp. 117–122.

[44] O. Ryhmä. *Ketterä toimintatapa*. 2019. URL: `https://www.op.fi/op-ryhma/tietoa-ryhmasta/op-lyhyesti/kettera-toimintatapa` (visited on 10/27/2019).

[45] K. Rautiainen, J. von Schantz, and J. Vahaniitty. “Supporting scaling agile with portfolio management: case Paf. com”. In: *System Sciences (HICSS), 2011 44th Hawaii International Conference on*. IEEE. 2011, pp. 1–10.

[46] J. Vähäniitty and K. Rautiainen. “Towards an Approach for Development Portfolio Management in Small Product-Oriented Software Companies”. In: *Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS-38)*. 2005, pp. 25–28.

[47] A. Shalloway, G. Beaver, and J. R. Trott. *Lean-Agile Software Development: Achieving Enterprise Agility*. Pearson Education, 2009.

[48] D. Leffingwell. *Scaling software agility: best practices for large enterprises*. Pearson Education, 2007.

[49] M. O. Ahmad, L. E. Lwakatare, P. Kuvaja, M. Oivo, and J. Markkula. “An empirical study of portfolio management and Kanban in agile and lean software companies”. In: *Journal of Software: Evolution and Process* 29.6 (2017), e1834.

[50] Atlassian. *Jira*. 2019. URL: `https://www.atlassian.com/software/jira` (visited on 02/25/2019).

[51] Atlassian. *Confluence*. 2019. URL: `https://www.atlassian.com/software/confluence` (visited on 02/25/2019).

[52] Tableau. *Tableau*. 2019. URL: `https://www.tableau.com` (visited on 02/25/2019).