Aalto University
School of Science
Degree Programme in Computer Science and Engineering

Oskar Ehnström

Lean software development and the effects of working with unfamiliar technology

A case study

Master's Thesis
Espoo, December 1, 2015

DRAFT! — October 6, 2015 — DRAFT!

Supervisor: Professor Marjo Kauppinen, Aalto University
Advisor: Suvi Uski (?)

Aalto University
School of Science
Degree Programme in Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author: Oskar Ehnström

Title:
Lean software development and the effects of working with unfamiliar technology
A case study

Date:	December 1, 2015	**Pages:**	21
Major:	Software Engineering and Business	**Code:**	T-76
Supervisor:	Professor Marjo Kauppinen		
Advisor:	Suvi Uski (?)		

This thesis presents the core concepts of lean software development and how they are affected by the introduction of unfamiliar technology. Existing case studies on lean software development projects are analyzed for practical implementations of the core principles. The case studies are compared to each other and to the principles presented in the seminal works on lean software development.

The current literature on lean software development is derived from the literature on lean manufacturing. Lean thinking was originally based on five principles: value, value stream, flow, pull and perfection. These have since been refined to seven principles specific to software development: optimize the whole, eliminate waste, build quality in, learn constantly, deliver fast, engage everyone and keep getting better. These principles can be applied as various practices in software projects depending on the context.

The principles hold well for the general domain. This thesis studies whether they work as such for projects where the technology domain is also unfamiliar to the project team. The study finds that the principles either hold or do not hold. The thesis speculates as to why they do or do not hold up as such.

Keywords: lean, lean software, lean software project, service creation, agile, LSC

Language: English

Aalto-yliopisto
Perustieteiden korkeakoulu
Tietotekniikan koulutusohjelma

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä:	Oskar Ehnström		
Työn nimi:			
Päiväys:	1. joulukuuta 2015	**Sivumäärä:**	21
Pääaine:	Ohjelmistotuotanto ja liiketoiminta	**Koodi:**	T-76
Valvoja:	Professori Marjo Kauppinen		
Ohjaaja:	Suvi Uski (?)		

Lorem ipsum Tempor tempor labore Ut reprehenderit tempor irure incididunt non labore irure dolore consectetur esse sit magna culpa ad consequat sit Ut velit veniam dolore fugiat sed nostrud reprehenderit.

Lorem ipsum Magna non sint incididunt laboris Ut proident exercitation dolore non eu adipisicing ullamco occaecat cupidatat Duis labore eiusmod veniam nisi dolor ea est deserunt exercitation aliquip commodo anim magna dolore anim ad mollit enim officia proident nisi quis.

Lorem ipsum Culpa enim sed tempor velit incididunt dolor aliqua consectetur ut quis officia consectetur proident magna sunt nulla cupidatat laboris et est nisi dolore exercitation consectetur est consequat ea aliqua officia Excepteur reprehenderit aliqua cillum.

Asiasanat:

Kieli: Englanti

Aalto-universitetet
Högskolan för teknikvetenskaper

Aalto-universitetet
Högskolan för teknikvetenskaper
Examensprogram för datateknik

SAMMANDRAG AV
DIPLOMARBETET

Utfört av:	Oskar Ehnström		
Arbetets namn:			
Datum:	Den 1 December 2015	**Sidantal:**	21
Huvudämne:	Programvaruproduktion och affärsverksamhet	**Kod:**	T-76
Övervakare:	Professor Marjo Kauppinen		
Handledare:	Suvi Uski (?)		

Lorem ipsum Tempor tempor labore Ut reprehenderit tempor irure incididunt non labore irure dolore consectetur esse sit magna culpa ad consequat sit Ut velit veniam dolore fugiat sed nostrud reprehenderit.

Lorem ipsum Magna non sint incididunt laboris Ut proident exercitation dolore non eu adipisicing ullamco occaecat cupidatat Duis labore eiusmod veniam nisi dolor ea est deserunt exercitation aliquip commodo anim magna dolore anim ad mollit enim officia proident nisi quis.

Lorem ipsum Culpa enim sed tempor velit incididunt dolor aliqua consectetur ut quis officia consectetur proident magna sunt nulla cupidatat laboris et est nisi dolore exercitation consectetur est consequat ea aliqua officia Excepteur reprehenderit aliqua cillum.

Nyckelord:

Språk: Engelska

Acknowledgements

TODO: Thank people here

Espoo, December 1, 2015

Oskar Ehnström

Contents

Chapter 1

Introduction

Lean thinking has been gaining traction in the software development industry in recent years. Books like "The Lean Startup" and "Lean Software Development: An Agile Toolkit" have taught us how to take lean practices that were initially used in manufacturing and use them to create value in software development and service design. Organizations have, at this point, had the time to implement lean practices and try the suggested methods in practice. This makes it possible to compare findings and analyze what the common best practices and problems are.

Although uncertainty is always a factor in software projects, new and unfamiliar technology introduces even more of it and can potentially lead to budget overruns or even project failure. Understanding how to adjust accepted lean practices to these projects would enable projects to perform more reliable estimates and avoid introducing unnecessary risk when striving for high rewards using new technology.

1.1 Scope

The scope of the literature review will be existing literature on lean software projects, comparing these to find similarities if there are any.

Scope of the empirical study will be one lean software development project. The project will be studied from the point of view of developers, customers and end-users. The thesis is limited to one case study.

1.2 Structure of the Thesis

This section presents the structure of the thesis.

!Fixme **Write these a bit smoother once the structure is done** Fixme!

Chapter 2 presents the origin of the lean philosophy. It goes through the development of lean principles from manufacturing to software development.

Chapter 3 covers the existing literature on lean software development. In this chapter experiences of previous lean software projects are analyzed and compared. This is done in order to find some common trends or best practices to use in projects. These common trends and best practices are compared in order to later compare them with the findings of the empirical study, which are presented in chapter 6.

Chapter 4 goes through the methods used in the empirical study. The participants and their roles are presented as well as the practical arrangements regarding interviews.

Chapter 5 presents the results of the empirical study. The data is analyzed and the results of that analysis are presented.

Chapter 6 discusses the results gathered from the empirical study and their implications. This is done by analyzing the relationship between the data gathered from the empirical study and the literature review presented int chapter 3.

Chapter 7 presents the conclusions of this thesis and suggestions for further research.

Chapter 2

Background

2.1 The origin of lean thinking

!Fixme **This section will be about the history of lean and how it came to be applied to software engineering** Fixme!

Even though the traditional wisdom is that the Japanese car manufacturers had a significant advantage over western competitors due to the lean methodologies they implemented there is some controversy over whether this was in fact the case. Dybá & Sharp argue that by examining the facts and taking automation into account the Japanese did not have a superior organizational advantage. [1]

2.2 Lean versus Agile

!Fixme **This section will discuss the differences between lean and agile** Fixme!

!Fixme **Lean should be thought of a set of principles rather than practices. This article has some excellent points and trends to talk about.[5]** Fixme!

Chapter 3

Literature review

This chapter presents the existing literature on lean software development. The methods used for gathering academic sources is presented, as well as a general overview on the subject. Finally existing literature on past lean software development projects are compared to the general principles of lean and to each other.

3.1 Literature overview

This section presents the methods used to gather material for the literature review.

Google Scholar was the primary source for material on lean and lean software development. Searching the numerous databases in Google Scholar with the keywords mentioned in figure 3.1. Once some suitable primary articles had been found these could be used to find related articles. By utilizing the related articles feature on Google Scholar more related articles could be found. The sources of known articles also led to other articles on the subject and especially to heavily cited articles that provide the foundation for many other articles. Searching for articles written by the authors of known articles also led to other articles that dealt with the subject. Conferences and journals that included some known articles also proved to include other similar articles and were a good source of material.

3.2 Lean software principles

!Fixme **In this section I will present the principles for lean software development as presented in [6] and subsequent articles.** Fixme!

Search term
lean software development
lean software management
lean project management
digital service creation
lean

Figure 3.1: Keywords used to find sources

3.3 Comparison of past case studies

This section compares the existing literature on lean software projects.

3.3.1 Timberline Inc.

Timberline Inc. case study. Probably the first case of adopting lean principles to software development.[3]

3.3.2 BBC Worldwide

BBC Worldwide case study. The gist of it is that the performance of the team improved when adopting lean practices, but there were some challenges in fitting the the lean principles with the rest of the company.[4]

3.3.3 Electrobit

Electrobit is a Finnish provider of wireless embedded systems. The case study was conducted in 2010 and the organization had used agile practices since 2007. The case study followed how some key performance indicators (KPI) changed as the organization started using lean practices. [8]

The study was particularly focused on how lean and agile practices can be combined in software development. The authors focused on elements that characterize the combination of lean and agile. They were also interested in what challenges the combination presents, as well as which elements of the combination were easy to implement.[8] As agile has only appeared to increase in popularity in the software development business this study is a good reference on how lean and agile can be combined in order to produce software. The hardware related business also presents some unique challenges compared to purely software based models of lean and agile.

The study found that the move to lean had indeed influenced practices of the company. Discussions with employees revealed that the lean principles had expanded concepts like reducing waste to be considered throughout the organization. In comparison, previous practices had left these responsibilities largely on the product owner alone. However, the change from agile to lean was described as "an incremental improvement in which Agile is not abandoned when Lean is adopted."[8]. This is quite natural, as agile practices focus more on the software development work whereas lean takes a more holistic approach to the whole value chain.

When the subject was speed and flexibility the discussions with the subjects of the study focused on the importance of short lead-times and the ability to cope with change.[8] These are important questions business-wise, as maneuvering fast and responding to change can be the deciding factor in a fast paced environment like software development. Responding to change is one of the things both agile and lean aim to enable in order to cope with chaotic systems.

Eliminating waste was seen as an aspect specifically related to lean principles that had not been incorporated in the earlier, agile, practices. Tightly related to that, seeing the whole also brought a more holistic approach to the company compared to the agile mindset before the change. When discussing specific practices, minimizing work-in-progress (WIP) was found to be easy to motivate and understand as a way to reduce waste.[8] This becomes clear when one thinks of unfinished code as being the software equivalent of inventory with possible bugs and unnecessary or unwanted features.

Electrobit's ways of working also enabled them to have short feedback cycles. This is a critical component for the ability to adapt and respond to change quickly. They also handled uncertainty by continuous learning. Estimations were small but accurate, which is the typical way for agile estimations to work. Another aspect that related to both agile and lean was that participants in the study pointed out that delaying decision making should not affect the release of the software.[8] This borrows from agile as it does not allow the schedule to be delayed, but recognizes that lean principles call for delaying a decision until it has to be made in order to keep all options open.

Lean principles emphasize "perfection" or "learn constantly", at Electrobit they found that Kanban provided an added value as a process because of its ability to visualize queues and enable finding the root cause of problems [8]. Agile also focuses on continuous improvement, so the change towards lean was likely a natural evolution towards organizational learning from the more team focused approach of agile.

Learning was enabled by organizational transparency. This was the most

stressed element of the discussions with the participants of the study. Transparency enabled knowledge sharing and enabled visibility on all organizational layers and in all directions. [8]

Finally, participants chose to mention the people factor of software development. This is very much related to agile, but has its place in lean principles as well in the form of "engage everyone" as presented by [6].

The study presents some challenges that Electrobit encountered regarding lean principles. Flexibility of the whole value stream was a challenge. Teams also perceived that they were unable remove waste even though they had identified it due to complex project set-ups. Long feedback loops were still an issue due to challenges in involving management and third parties into the development process. [8] These challenges are familiar from traditional lean manufacturing and agile software development, which could mean that lean software development was not a silver bullet in this case, even if it did improve the overall situation.

!Fixme **TODO: compare more thoroughly with other cases and poppendieck** Fixme!

3.3.4 Two Case Studies

In "Lean Software Development: Two Case Studies"[2] author Peter Middleton sets out to study whether lean principles can be applied to software development. This study appears to be one of the first studies that tackles this question and predates the work of Poppendieck & Poppendieck published in 2003.

The study presents the foundation of lean principles and how they might be applied to software development. The most important result, however, is the experiment conducted by Middleton to study the effects of applying lean methods on a traditional software process. In this experiment, two small teams in a large organization were selected to try lean practices in their daily work.[2]

The traditional process was first streamlined somewhat to enable the introduction of lean principles. Once the new process was stable, lean principles were introduced. The implementation focused most on aiming to reduce waste by stopping the process once a defect was found. This initially slowed down both teams, as team members were not allowed to work on other tasks while the issue was resolved. This was done to limit their WIP. Once the teams learned to see defects and address them more quickly overall progress improved compared to the traditional process.[2]

Although the study is very limited, focusing on two small teams for a short period of time, it is able to highlight some of the most prominent

organizational challenges. In this particular organization the hierarchical structure of the teams introduced friction in reporting issues, which is an essential part of lean. The hierarchy also fostered a culture where people needed to take jobs they had little aptitude for in order to advance in their careers. One manager also felt that they were securing their job by not sharing information. This could have been a result of the organizations less than ideal policies on learning, which were mentioned as a challenge for the application of lean. Finally, some aspects of the lean process were hindered by third parties inside the organization who were unable to deliver the required quality.[2]

An overall conclusion of the problems uncovered in the study was that problems were often a result of organizational challenges, and the issues with quality were, in fact, the symptom of deeper problems. Finally, the study also concluded that "no inherent reason has been found to suggest that lean techniques cannot be used in software process."[2]. This might have influenced others to try to replicate these results in larger studies.

!Fixme **TODO: compare more thoroughly with other cases and poppendieck** Fixme!

3.3.5 Others

3.4 Research problem and question

This chapter will end with the research problem and questions.

The research problem is defined as follows:

What are the commonly accepted and used lean software development practices and how do they change when working with new and unfamiliar technology?

To investigate this problem three research questions have been set up in table 3.1.

Question	Literature review	Empirical study
What are the currently available best practices for lean software projects?	x	
Which of these best practices need to be adapted when working with new technology?		x
How do these best practices need to adapted?		x
Which best practices remain valid?		x

Table 3.1: Research questions and their respective sections

Chapter 4

Methods

In this section I will describe how I conducted the empirical study.

- Interviews
  - 2-3 project members
  - 2-3 customers
  - 2-3 end-users
  - 1-1.5 hours each (5-10min for end-users)
- Semi-structured interviews

Chapter 5

Results

In this section I will present the results of the empirical study. This section contains the raw data of the conducted interviews. This section does not analyze or compare the results with the literature.

Chapter 6

Discussion

Here I will discuss how the findings from my empirical work relate to the literary review. What are the similarities and differences when comparing what the literature says and what the interviews showed.

6.1 Lean Service Creation

!Fixme **Lean Service Creation is about build measure learn and create something new. Tie this into the problem and focus on the new part. Using new devices maybe?** Fixme!

Lean Service Creation is based on the ideas of "The Lean Startup" as described by Eric Reis in his 2011 book.[7]

!Fixme **How do I get a source for this? Interview?** Fixme!

Chapter 7

Conclusions

Here I mention the most important findings of the discussion section and the literary review section.

I also point out how this research can be used in the future and what it's limitations are. (e.g. only one case study)

2 pages

Bibliography

[1] Dybà, T., and Sharp, H. What's the evidence for lean? *IEEE Software 29*, 5 (2012), 19–21.

[2] Middleton, P. Lean software development: Two case studies. *Software Quality Journal 9*, 4 (2001), 241–252.

[3] Middleton, P., Flaxel, A., and Cookson, A. Lean software management case study: Timberline inc. *Extreme Programming and Agile Processes in Software Engineering* (2005), 1–9.

[4] Middleton, P., and Joyce, D. Lean software management: Bbc worldwide case study. *IEEE Transactions on Engineering Management 59*, 1 (2012), 20–32.

[5] Poppendieck, M., and Cusumano, M. a. Lean software development: A tutorial. *IEEE Software 29*, 5 (2012), 26–32.

[6] Poppendieck, M., and Poppendieck, T. *Lean software development: an agile toolkit*. Addison-Wesley Professional, 2003.

[7] Ries, E. *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Random House LLC, 2011.

[8] Rodriguez, P., Partanen, J., Kuvaja, P., and Oivo, M. Combining lean thinking and agile methods for software development: A case study of a finnish provider of wireless embedded systems detailed. *2014 47th Hawaii International Conference on System Sciences* (2014), 4770–4779.

Appendix A

Interview questions

Here goes the questions from the interviews.