

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:	18
Major:	Software Engineering and Business	Code:	T-76
Supervisor:	Professor Marjo Kauppinen		
Advisor:	Suvi Uski (?)		
<p>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.</p> <p>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.</p> <p>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.</p>			
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:	18
Pääaine:	Ohjelmistotuotanto ja liiketoiminta	Koodi:	T-76
Valvoja:	Professori Marjo Kauppinen		
Ohjaaja:	Suvi Uski (?)		
<p>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.</p> <p>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.</p> <p>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.</p>			
Asiasanat:			
Kieli:	Englanti		

Aalto-universitetet
Högskolan för teknikvetenskaper
Examensprogram för datateknik

SAMMANDRAG AV
DIPLOMARBETET

Utfört av:	Oskar Ehnström		
Arbetets namn:			
Datum:	Den 1 December 2015	Sidantal:	18
Huvudämne:	Programvaruproduktion och affärsverksamhet	Kod:	T-76
Övervakare:	Professor Marjo Kauppinen		
Handledare:	Suvi Uski (?)		
<p>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.</p> <p>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.</p> <p>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.</p>			
Nyckelord:			
Språk:	Engelska		

Acknowledgements

TODO: Thank people here

Espoo, December 1, 2015

Oskar Ehnström

Contents

1	Introduction	7
1.1	Scope	7
1.2	Structure of the Thesis	7
2	Background	9
2.1	The origin of lean thinking	9
2.2	Lean versus Agile	9
3	Literature review	10
3.1	Literature overview	10
3.2	Lean software principles	10
3.3	Comparison of past lean projects	11
3.4	Research problem and question	11
4	Methods	13
5	Results	14
6	Discussion	15
6.1	Lean Service Creation	15
7	Conclusions	16
A	Interview questions	18

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 in 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 lean projects

In this section I will compare articles regarding case studies focusing on lean software development/management.

Two case studies. This is an early attempt to test lean software development. The article covers an experiment set up to test the validity of using lean principles in software development. [2]

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

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]

3.4 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 re- view	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 be 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.

Appendix A

Interview questions

Here goes the questions from the interviews.