

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 2, 2015 — DRAFT!

Supervisor:	Professor Marjo Kauppinen, Aalto University
Advisor:	Eeva Suvi

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:	17
Major:	Software Engineering and Business	Code:	T-76
Supervisor:	Professor Marjo Kauppinen		
Advisor:	Eeva Suvi		
<p>A dissertation or thesis is a document submitted in support of candidature for a degree or professional qualification presenting the author's research and findings. In some countries/universities, the word thesis or a cognate is used as part of a bachelor's or master's course, while dissertation is normally applied to a doctorate, whilst, in others, the reverse is true.</p> <p>!FIXME Abstract text goes here (and this is an example how to use fixme). FIXME! Fixme is a command that helps you identify parts of your thesis that still require some work. When compiled in the custom <code>mydraft</code> mode, text parts tagged with <code>fixmes</code> are shown in bold and with <code>fixme</code> tags around them. When compiled in normal mode, the <code>fixme</code>-tagged text is shown normally (without special formatting). The draft mode also causes the "Draft" text to appear on the front page, alongside with the document compilation date. The custom <code>mydraft</code> mode is selected by the <code>mydraft</code> option given for the package <code>aalto-thesis</code>, near the top of the <code>thesis-example.tex</code> file.</p> <p>The thesis example file (<code>thesis-example.tex</code>), all the chapter content files (<code>1introduction.tex</code> and so on), and the Aalto style file (<code>aalto-thesis.sty</code>) are commented with explanations on how the Aalto thesis works. The files also contain some examples on how to customize various details of the thesis layout, and of course the example text works as an example in itself. Please read the comments and the example text; that should get you well on your way!</p>			
Keywords:	innovation, services, service design		
Language:	English		

Aalto-yliopisto
 Perustieteiden korkeakoulu
 Tietotekniikan koulutusohjelma

DIPLOMITYÖN
 TIIVISTELMÄ

Tekijä:	Oskar Ehnström		
Työn nimi:			
Päiväys:	1. joulukuuta 2015	Sivumäärä:	17
Pääaine:	Ohjelmistotuotanto ja liiketoiminta	Koodi:	T-76
Valvoja:	Professori Marjo Kauppinen		
Ohjaaja:	Eeva Suvi		
<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:	innovaatio, palvelut, palvelusuunnittelu		
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:	17
Huvudämne:	Programvaruproduktion och affärsverksamhet	Kod:	T-76
Övervakare:	Professor Marjo Kauppinen		
Handledare:	Eeva Suvi		
<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:	innovation, tjänster, service design		
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	Litterature review	10
3.1	Litterature overview	10
3.2	Lean software principles	11
3.3	Comparison of past lean projects	11
3.4	Research problem and question	11
4	Methods	12
5	Results	13
6	Discussion	14
6.1	Lean Service Creation	14
7	Conclusions	15
A	Interview questions	17

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. Organisations 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 analyse 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 analysed 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 analysing 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 controcersy over wether 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

Litterature review

In this section I will present the existing litterature and what general guidelines or conclusions it presents. I will refer to the main concepts presented in books and compare those findings to case studies that have implemented the relevant concepts in order to find some commonalities if there are any.

I will also present the process for finding and analyzing the material I have chosen.

3.1 Litterature overview

- How was the litterature review conducted?
 - Scholar
 - Scholar through related articles of known good articles
 - Key articles then their sources
 - Same author, look for more
 - Same conference/journal, look for more

Keywords: *lean software development, lean software management, lean project management, digital service creation, service-dominant, design thinking, new service development*

Question	Litterature 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 be adapted?		x
Which best practices remain valid?		x

Table 3.1: Research questions and their respective sections

3.2 Lean software principles

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

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 litterature.

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.[6]

!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 its limitations are. (e.g. only one case study)

2 pages

Bibliography

- [1] DYBÅ , 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] 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.