Lean Software Development

Mary Poppendieck
Poppendieck.LLC
mary@poppendieck.com

Abstract

Lean Software Development is the application of the principles of the Toyota Product Development System to software development. Toyota has been extremely successful developing complex new vehicles, which include a vast amount of embedded software, in a very short time and always on time. This tutorial examines the underlying engineering principles Toyota uses to develop vehicles and shows how they can be applied to software development. When correctly applied, lean software development results in high quality software that is developed quickly and at the lowest possible cost. Moreover, the success of many of the practices of Agile Software Development can be explained by understanding the principles of Lean Software development.

1. Introduction

As global competitiveness comes to the software development industry, the search is on for a better way to create first-class software rapidly, repeatedly, and reliably. Lean initiatives in manufacturing, logistics, services, and product development have led to dramatic improvements in cost, quality and delivery time; can they do the same for software development? The short answer is "Absolutely!"

Of the many methods that have arisen to improve software engineering processes, Lean is emerging as one that is grounded in decades of work understanding how to make processes better. Lean thinking focuses on giving customers what they want, when and where the want it, without a wasted motion or wasted minute.

This one day tutorial provides an introduction to applying lean principles to software development. It focuses on these seven lean principles:

1. Eliminate Waste
2. Build Quality In
3. Create Knowledge
4. Defer Commitment
5. Deliver Fast
6. Respect People
7. Optimize the Whole

As a group, these principles provide guidance on how to deliver software faster, better, and for less cost – all at the same time.

2. Summary of Contents

The following is an outline of a one day tutorial.

2.1. History of Lean

This section discusses the pillars of the Toyota Production System: Just-in-Time Flow and Stop-the-Line quality. It discusses the charter of a QA function in a lean environment and how all of these concepts apply directly to software development.

2.2. Waste

This section discusses the fundamental lean concept of waste and how to look for waste in a development system (as opposed to a production system). It covers the seven wastes of software development.

The class will do a value stream mapping exercise: groups of about seven people will draw a value stream map representing the actual experience of one person in the group. Several groups will present their map to the class, and it will be critiqued by the presenters.

2.3. Quality

This section will present the case for test-driven development, automated unit and acceptance testing, continuous integration, and nested synchronization. It will be focused not so much on how to do it as why it is fundamental in a lean environment.

29th International Conference on Software Engineering (ICSE'07 Companion)
0-7695-2892-9/07 $20.00 © 2007 IEEE

This section will also cover how to achieve regulatory compliance (FDA, SOX, etc.) using the recommended approaches.

2.4. Knowledge

This section will cover the rationale behind delaying commitment and present methods of preserving knowledge so that teams do not have to re-learn what was once known. It will discuss the benefits of set-based design, a counter-intuitive approach which involves exploring multiple options instead of focusing on a single solution.

2.5. Speed

This section will discuss the application of queuing theory to software development. It will demonstrate why long queues are detrimental to effective development and show how to avoid them. It will also show why rapid cycle time leads to higher quality and reduced cost.

2.6. System

This section will discuss why it is not sufficient to simply focus on software development, and show how to shift the focus of development to the overall objective of the effort: the product or process supported by the software. In addition, the section will demonstrate how decomposing measurements leads to sub-optimization, and propose aggregate measurements which have been shown to drive effective behavior.

3. Why Lean is Important

Lean Software Development provides the theory behind agile software development practices and gives organizations a set of principles from which to fashion software engineering processes that will work best in the context of their customers, their domain, their development capability, and their unique situation.

Mary and Tom Poppendieck have taught dozens of classes on Lean Software Development and have written two books explaining the material they teach in their classes. These classes and books have led many organizations to reexamine and restructure software engineering practices, resulting in dramatic improvements in the quality, cost, and development speed of products and processes containing software.

4. Presenters

This tutorial will be presented by Mary Poppendieck and Tom Poppendieck.

4.1. Mary Poppendieck

Mary Poppendieck has been in the Information Technology industry for over thirty years. She has managed software development, supply chain management, manufacturing operations, and new product development. She spearheaded the implementation of a Just-in-Time system in a 3M video tape manufacturing plant and led new product development teams, commercializing products ranging from digital controllers to 3M Light Fiber™.

Mary is a popular writer and speaker, and coauthor of the book *Lean Software Development* [1], which was awarded the Software Development Productivity Award in 2004. A sequel, *Implementing Lean Software Development* [2], was recently published.

4.2. Tom Poppendieck

Tom Poppendieck has 25 years of experience in computing including eight years of work with object technology. His modeling and mentoring skills are rooted in his experience as a physics professor. His early work was in IT infrastructure, product development, and manufacturing support, and evolved to consulting project assignments in healthcare, logistics, mortgage banking, and travel services.

Tom holds a PhD in Physics and has taught physics for ten years. He is coauthor of the book *Lean Software Development* [1], which was awarded the Software Development Productivity Award in 2004. A sequel, *Implementing Lean Software Development* [2], was recently published.