

Implementing Lean Software Development From Concept to Cash

The Sequel

Lean was an idea borrowed from the 1990's when we wrote the book ***Lean Software Development: An Agile Toolkit*** in 2003. We had observed that breakthrough ideas from manufacturing and logistics often take a decade or two before they are adapted to provide suitable guidance for development efforts. So we decided it was not too late to use well-proven lean concepts from the 1980's and 1990's to help us explain why agile methods are a very effective approach to software development.

The strategy worked. ***Lean Software Development*** presents a set of thinking tools based on lean thinking that leaders continue to find useful for understanding agile software development. The book has been purchased by many a developer who gave it to his or her manager to read, and many managers have distributed multiple copies of the book to colleagues in support of a transition to lean/agile software development.

Meanwhile, something unexpected happened to *lean*. In the last couple of years lean initiatives have experienced a resurgence in popularity. The word *lean* was originally popularized in the early 1990's to characterize the Japanese approach to automobile manufacturing.¹ In recent years, Honda and Toyota have been doing increasingly well in the North American auto market, while Detroit automakers are restructuring. For example, Toyota's profits rose from over \$8 billion in the fiscal year ending March 31, 2003 to over \$10 billion in 2004, \$11 billion in 2005, and \$12 billion in 2006. Many companies have taken a second look at *lean* to try to understand what's behind such steady and sustained success.

Lean initiatives seldom start in the software development or product development part of a company, but over time, successful lean initiatives make their way from manufacturing or logistics to development departments. However, lean practices from manufacturing and other operational areas do not adapt easily to a development environment, so lean initiatives have a tendency to stall when they reach software development. While the underlying lean principles remain valid, it is usually inappropriate to apply operational practices and measurements to a development environment. When lean initiatives stall in software development areas, many companies have discovered that ***Lean Software Development*** gives them a good foundation for thinking about how to modify their approach and adapt lean ideas to a development organization.

¹ James Womack, Daniel Jones and Daniel Roos, ***The Machine that Changed the World***, Rawson Associates, 1990.

The benefits of lean and agile software development have become widely known and appreciated in the last couple of years, and many organizations are changing the way they develop software. We have traveled around the world visiting organizations as they implement these new approaches, and we have learned a lot from our interaction with people working hard to change the way they develop software. As our knowledge has grown, so has the demand for more information on implementing lean software development. We realized that a book would allow us to share what we've learned with many more people than we can contact personally. Therefore we have summarized our experiences in this book: ***Implementing Lean Software Development: From Concept to Cash.***

This book is not a cookbook for implementing lean software development; like our last book, it is a set of thinking tools about how to go about adapting lean principles to your world. We start this book where the last book left off and go deeper into the issues and problems that people encounter when trying to implement lean and agile software development. You might consider this book a sequel to ***Lean Software Development***; instead of repeating what is in that book, we take a different perspective. We assume the reader is convinced that lean/agile software development is a good idea, and focus on the essential elements of a successful implementation. We look at key aspects of implementation and discuss what is important, what isn't, and why. Our objective is to help organizations get started down the path toward more effective software development.

The first chapter of this book reviews the history of *lean*, and the second chapter reviews the seven principles of lean software development presented in ***Lean Software Development***. These are followed by chapters on *value*, *waste*, *speed*, *people*, *knowledge*, *quality*, *partners*, and the *journey* ahead. Each of these eight chapters begins with a story that illustrates how one organization dealt with the issue at hand. This is followed by a discussion of key topics we have found to be important, along with short stories that illustrate the topic and answers to typical questions we often hear. Each chapter ends with a set of exercises that helps you explore the topics more deeply.

May the wind be always at your back.

Mary and Tom Poppendieck

May, 2006