



Foreword

Jeff Sutherland

I created the first Scrum in 1993 at Easel Corporation in Burlington, Massachusetts,¹ in cooperation with our CEO, who bet his company on the team that made the first Scrum work. In 1995, I began working with Ken Schwaber, who formalized the process for worldwide deployment.² In four companies since Easel, I have used the self-organizing nature of Scrum to move from a Type A Scrum (winning teams) to a Type B Scrum (winning product portfolios) to a Type C Scrum (winning companies).

In 2000, I introduced Scrum to PatientKeeper, and this has proven to be an excellent vehicle for lean development. Over the years we have shortened cycle times to exactly what customers need: one week for critical items, one month for minor enhancements, and three months for significant new products. The three-month releases are accumulations of one-month Sprints, and every Sprint is a release.

At PatientKeeper the entire company runs as a Scrum and inspects, adapts, self-organizes, and changes every Monday when we have our MetaScrum meeting.³ The Product Owner runs this meeting, and all company stakeholders are present, including the CEO. Lean concepts are carefully examined here, and Sprints are started, stopped, or changed only in this meeting. The whole company, including affected customers, can be reset in one afternoon with decisions made during the weekly MetaScrum.

-
1. Sutherland, J., "Agile Development: Lessons Learned from the First Scrum," Cutter Agile Project Management Advisory Service: Executive Update, 2004, 5(20): pp. 1–4.
 2. Schwaber, K., "Scrum Development Process," in *OOPSLA Business Object Design and Implementation Workshop*, J. Sutherland, et al., editors. 1997, Springer: London.
 3. Sutherland, J., "Future of Scrum: Parallel Pipelining of Sprints in Complex Projects," in *AGILE 2005 Conference*. 2005. Denver, Colorado: IEEE.

We have a chief Product Owner with a team that gets the Product Backlog “ready.” We have a chief ScrumMaster, who runs a 15-minute daily Scrum of Scrums. What did teams do yesterday, what will they do today, and what are the impediments to delivery? We manage 45 releases of software per year into large, mission-critical enterprises using these short meetings and using an automated system that tracks the state of concurrent Sprint backlogs. Dates are committed to customers before the start of a Sprint, and thousands of Web users and hundreds of physicians with PDAs go live at the end of every Sprint. For PatientKeeper, concept (Product Backlog) to cash (product in production) occurs in one-month intervals. We have had to eliminate waste everywhere—before, during, and after the implementation process.

In 1993, lean product development was not well understood in any industry. Scrum was the first concrete implementation of lean thinking to software development that allowed organizations of all types and sizes to start up lean teams in a couple of days using a standard pattern that was easily understood. What was hard was explaining why and how to implement the pattern to generate continuous quality and productivity improvement.

Today, the writing and courses from Mary and Tom Poppendieck provide a proven set of principles that organizations can use to adapt tools, techniques, and methods to their own specific unique contexts and capabilities. Now we can explain how to use Scrum to lean out software development. In addition to my company, PatientKeeper, I use the procedures and processes outlined in this book to teach practitioners worldwide how to optimize Scrum.

Lean software development views all agile methods as valid, proven applications of lean thinking to software. It also goes beyond agile, providing a broader perspective that enables agile methods to thrive. First, it looks along the whole value chain, from concept to cash and tries to address all the waste and delays that happen before and after the coders contribute their part. Second, it establishes a management context to extend, nourish, and leverage agile software practices. Third, it provides a proven set of principles that each organization can use to adapt tools, techniques, and methods to their own specific unique contexts and capabilities.

Every chapter in this book illustrates a set of principles that can be implemented to build more productive teams. If you want to be like Toyota, where productivity is consistently four times that of its competitors and quality is twelve times better, these practices are essential. If you execute well on the principles in this book, resistance by your competition is futile, and winning in your market is certain. The return on investment in practices outlined in this book can be very high.

To create the lean “secret sauce” for your company, we have found you must systematically implement this accumulated lean wisdom. In short, you must deeply understand the Japanese concepts of Muri, Mura, and Muda. Mary and Tom unfold them as the seven lean principles and the seven wastes of software development to help you easily understand how they work and know what to do.

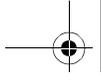
Muri relates to properly loading a system and Mura relates to never stressing a person, system, or process. Yet many managers want to load developers at 110 percent. They desperately want to create a greater sense of “urgency” so developers will “work harder.” They want to micromanage teams, which stifles self-organization. These ill-conceived notions often introduce wait time, churn, death marches, burnout, and failed projects.

When I ask technical managers whether they load the CPU on their laptop to 110 percent they laugh and say, “Of course not. My computer would stop running!” Yet by overloading teams, projects are often late, software is brittle and hard to maintain, and things gradually get worse, not better. One needs to understand that Toyota and Scrum use a pull system that avoids stress (Mura) and eliminates bottlenecks (Muri). The developers take what they need off the Product Backlog just in time. They choose only Product Backlog that is ready to take into a Sprint, and they never take more than they can get done in a Sprint. They go faster by surfacing impediments and working with management to eliminate waste (Muda). By removing waste, they have less work to do. Productivity goes up, and they have time to deliver quality software and focus on exactly what the customers need.

So by understanding loading (Muri) and avoiding stress (Mura) you shorten cycle time. This leans out the environment causing impediments that create waste (Muda) to be highly visible. When you eliminate these impediments, your teams move faster, do more with less, increase quality, and move the product right into the sweet spot for the customer.

I recommend you keep the Poppendiecks’ books on your desk and use them regularly to help with systematic and continuous implementation of lean principles. Practice hard, and you will rapidly double productivity by using lean development to do less by avoiding waste, and then double it again by working smarter by eliminating impediments. By going four times faster in the right way, your quality will improve by a factor of twelve like Toyota.

—Jeff Sutherland, Ph.D.
Chief Technology Officer, PatientKeeper
Certified ScrumMaster Trainer
Inventor of the Scrum Development Process
July 2006



Foreword

Kent Beck

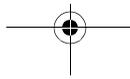
Software development is a chain with many links, and improving overall effectiveness requires looking at the whole chain. This book is for people involved in software development—not just programmers, but also managers, sponsors, customers, testers, and designers. The principles presented here eventually affect everyone connected to the development of software. This book speaks to those of you who are ready to look at the big picture of development.

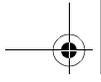
For ideas to have impact, they must be grounded in both theory and experience. By translating well-proven ideas from lean manufacturing, this book presents theory and practice and applies them to software development. Many of the fundamental problems with manufacturing are also problems with software development: dealing with uncertainty and change, continuously improving processes, and delivering value to customers.

What does this book offer you? First, a wide variety of theories that provide alternative ways of thinking about how software development can be improved. Second, a set of stories about the application of these theories to real projects. Third, provocative questions to help you apply the theories and the lessons in the stories to your unique situation.

Mary is uniquely qualified to present this material. She experienced lean manufacturing firsthand, helping rescue a plant from being shut down by transforming its operations. She's also experienced software development as a programmer and a manager. I once shared a seminar with Mary, and it was a treat to see the force and confidence with which she presents her material. I can hear that same direct voice in her writing here.

If you've read the Poppendiecks' precursor book, *Lean Software Development*, this book offers new guidance towards its implementation. It reiterates the theory in the previous volume, but always with an eye towards applying the ideas in real situations. The result is a readable and practical guide to ideas that have the potential to help you transform your development.





FOREWORD



This is a book for doers, doers who want their actions aligned with where they can do the most good. If you are such a person, I recommend this book to you as a source of ideas and energy for positive change.

—Kent Beck
Three Rivers Institute
July 2006

