



H3RALD

Book Review: Leading Lean Software Development

A lean leadership framework

by *Fabio Cevasco*

February 2013

Originally published on H3RALD.com

Structure and Organization	2
Chapter 1: Systems Thinking	2
Chapter 2: Technical Excellence.....	3
Chapter 3: Reliable Delivery.....	3
Chapter 4: Relentless Improvement	3
Chapter 5: Great People	3
Chapter 6: Aligned Leaders	4
Final Thoughts.....	4

If you already heard the names Mary and Tom Poppendieck, chances are that you already know what *Lean Software Development* is. If you don't, start from [this Wikipedia page](#). Mary and Tom coined this term with their first book on the subject [Lean Software Development: An Agile Toolkit](#), that was followed three years later by [Implementing Lean Software Development: From Concept to Cash](#), and finally by this book: [Leading Lean Software Development: Results Are not the Point](#).

Unlike the two other books, this one is focused about making lean software practices succeed. In some way, it can be compared to [Succeeding with Agile](#), but while Mike Cohn's book focuses entirely on Scrum, this book has a much broader scope. Moreover, the book contains a lot of digressions and stories—even not directly related to software development—aimed at understanding particular aspects of Lean Software Development and the Lean movement in general.

The focus is, as the title suggests, on leadership: how can you be a good leader in these difficult, ever-changing times? How can you be agile without losing your team? How can you improve the existing processes so that they can help you achieve your goals? If you ever asked yourself these questions, this is the right book for you...

Structure and Organization

This book is extremely well-structured. Its Table of Contents follows some very rigid rules which make this book one of the most organized texts I've ever come across. It is divided into six chapters, each organized as follows:

- A *snapshot* or an introductory story for the chapter's main topic
- Four *frames*, each describing a lean practice or personal quality
- A *portrait* of a leader
- *Your Shot*, i.e. some questions and exercises for the readers

In total, the book contains 24 frames constituting the “Big Picture”, which is actually a very powerful framework for lean software leadership. You can read the book's TOC [online](#) on the Poppendieck website and read the book's Introduction ([PDF link](#)) on the whole concept of *framing* (yes, both the authors do love photography!). When I started my career as a technical writer I used to love carefully-structured, symmetrical manuals. After a while, however, I understood that such rigorous structuring can even be dangerous if it becomes an obsession: you end up adding extra “padding writing” to make sections roughly match in length, or you start cutting down some other parts, for the same reason. Writing well-balanced books is hard, but I must say that the authors do a very good job with this book: it flows very naturally while keeping to its rigorous structure.



Chapter 1: Systems Thinking

The first chapter is about customers, what they want and the goals of your system. It describes some interesting high level concepts like *failure demand* and *policy-driven waste*, and how to spot opportunities to improve the process.

What I found particularly interesting was the usage of [process maps](#) to analyze an existing process and find bottlenecks or leaks (in terms of time). I was instantly sold on this practice after reading the success story of how a company manage to reduce the overall time to process and solve customer issues simply by connecting customers directly to developers instead of tech support engineers. This is something you can't apply everywhere, but after creating a process map for that specific case, the solution was evident.

More generally speaking, this chapter provides a recipe/checklist outlining the sequence of the phases of process improvement and problem solving:

1. *Understand*
2. *Observe*
3. *Visualize*
4. *Evaluate*
5. *Implement*

Chapter 2: Technical Excellence

This is the only chapter focusing primarily on technical topics and knowledge. It starts with a very lengthy digression on the history of programming methodologies, aimed at understanding *what works and what doesn't*. Some examples of IT stuff that worked include the Internet, PCs and ...Open Source Software.

This chapter provides a general overview on Software Development as a whole. It contains some interesting information on software complexity and dealing with architectural dependencies, comprehensive sections on testing and continuous integration, and just a half page on refactoring (understandable, seeing that there are already plenty of excellent books on the subject).

Chapter 3: Reliable Delivery

The *Race to the Sky* section at the beginning of Chapter 3 is by far the most fascinating of the non-IT stories included in this book. It describes the construction of the Empire State Building in 1930, how it was planned out, what strategies were followed, and why it succeeded (why *the construction* succeeded: the building itself remained totally unprofitable for quite some time).

There are [plenty](#) of books on the subject, but Tom and Mary Poppendieck well summarize the key points of this modern-day epic achievement: how to build the tallest skyscraper in the world in a single year. This story teaches us how to work under very tight deadlines, by designing a system to fit constraints, rather than estimating up-front.

This story was perfect to introduce, in the same chapter, concepts like [Kanban](#), *pull scheduling* and *adaptive control*, which only recently have been seriously considered in the world of Software Development but they are becoming more and more relevant.

Chapter 4: Relentless Improvement

Chapter 4 starts with a brief history of the checklist, which was invented in 1935, to be used by airplane pilots. It then moves on to its usage in hospitals, describing how checklists helped dropping infections caused by inserting central venous catheters incorrectly. Why all this? To focus on the concept of *process standards*, or better, how *we* can improve processes to accomplish our goals.

Basically, this is what Toyota does: regulations should not be written on stone, but they should be reviewed and updated frequently for continuous improvement or [Kaizen](#).

Finally, this chapter also briefly introduces a few different ways to perform root-cause analysis, such as using [fishbone diagrams](#).

Chapter 5: Great People

This chapter and the last one are actually focused on people and management. In this chapter, an unusual (for this kind of books, that is) but intriguing analysis on different countries using the following dimensions:

- power distance
- individualism
- masculinity
- uncertainty avoidance
- long-term orientation

Turns out that individualism is abundant in the Western world but not so much in the Far East (who would have thought!), but the opposite applies to power distance. A bit stereotypical, if I may, but not too much: the results are not surprising, especially when it comes to considering different cultures as a whole. Once more, the focus is again on Toyota's Kaizen and their culture of *respect for the people*.

On page 198, the meaning of the subtitle of the book (Results Are not the Point) is revealed: "developing the people and the system so that together they are capable to achieve successful results is the point". Agile is precisely about this: focusing on the people.

But what about leaders? This is an aspect of the whole Agile philosophy that I keep stumbling upon: if you want *The Team* to be in charge, what happens to leadership? As I found out myself working in and with Agile Teams, often there's a serious lack of strong leaders. "Leadership needs to be gently refactored into Agile", that's what Mary and Tom recommend. How? It depends on each specific case, but it must always be done *gently*.

Chapter 6: Aligned Leaders

The final chapter begins with the history of *Agile@IBM*, or how to turn the biggest software company in the world into a massive agile machine. It wasn't a top-down decision, the CEO didn't just wake up one morning and decided that everyone should go Agile. Quite the opposite: it was something that was *pulled* by developers rather than *pushed* at them.

In cases like this, companies should be focusing on developing people, including good leaders, instead of particular initiatives and processes. Leaders in turn should shift their focus from details to more high-level decisions. When it comes to facing changes, leaders should look at the [12 principles](#) of the [BBRT](#) leadership model.

The final portrait, *Leaders at all Levels*, well summarizes the key to successful leadership: "leadership is about example, coaching and helping others to achieve their goals".

Final Thoughts

If you're looking for a manual on implementing Lean Software Development in detail, this is probably not the best book on the subject. If you're a developer at the start of your career, with no management responsibilities, you'd want more technical juice, so probably you should read the other two books by the Poppendiecks on the subject first.

On the other hand, if you have been working in IT for a few years, and maybe you already started to climb up the corporate ladder, reading this book could make the difference between being successful leader or not. This book does not go very in-depth with any particular methodology or process, but it does provide an excellent overview of a lot of them.

To get the best out of *Leading Lean Software Development*, you should read it a least once sequentially, skipping the parts that are not relevant to you (right now), taking notes on the more interesting frames, and then go back over them to digest them properly. Do *not* skip the introduction of each chapter though, for one because they are always pleasant to read between frames, and also because they do teach some very important values or strategies that you *must* assimilate.

Book Review: Leading Lean Software Development - A lean leadership framework

The general message that stands out when reading this book is *focus on people*. Customers, of course, but also employees: every single successful company mentioned in this book, from Toyota to Southwest Airlines, became successful because they always focused on developing people *first*, and *then* products.