Lean Software Development

Lean Software Development
An Agile Toolkit

The Agile Software Development Se

Cockburn • Highsmith
Series Editors

- Adapting agile practices to your development organization
- Uncovering and eradicating waste throughout the software development lifecycle
- Practical techniques for every development manager, project manager, and technical leader

Forewords by
Jim Highsmith and Ken Schwaber

Mary Poppendiec
Tom Poppendieck

The Addison-Wesley Signature Series

IMPLEMENTING LEAN SOFTWARE DEVELOPMENT

FROM CONCEPT TO CASH

MARY AND TOM POPPENDIECK

Forewords by Jeff Sutherland and Kent Beck

The Addison-Wesley Signature Series

LEADING LEAN SOFTWARE DEVELOPMENT

RESULTS ARE NOT THE POINT

MARY AND TOM POPPENDIECK

Foreword by Dottie Acton,
Senior Fellow, Lockheed Martin

09/12/12

History

• Paris, France, July 1785. Honoré Blanc invented interchangeable parts for building guns. It was 50 firing mechanisms (called "locks") packaged in a box.

• Detroit, USA, January 1914. Henry Ford raised wages of workers from $2.40 for a nine-hour day to $5 for an eight-hour day as he began assembly line production of the Model T.

• Koromo, Japan, October 1949. Passenger car production restrictions were lifted in post-war Japan. In 1945, Kiichiro Toyoda had challenged his company to "catch up with America," but it was clear that Toyota could not catch up by adopting America's mass production model. Instead they went for "Just-in-Time" assembly (parts should be manufacture right before they are needed). Which lead to companywide adoption of the Toyota Production System.

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**

Principle 1: Eliminate Waste

Manufacturing	Software Development
In-Process Inventory	Partially Done Work
Over-Production	Extra Features
Extra Processing	Relearning
Transportation	Handoffs
Motion	Task Switching
Waiting	Delays
Defects	Defects

The Seven Wastes

09/12/12

Principle 2: Build Quality in

Practices:

- TDD
- Continuos Integration
- Exploratory testing
- Pair programming
- Code reviews

Principle 3: Create Knowledge

Alan MacCormack has identified four practices that lead to successful software development:

1. Early release of a minimum feature set to customers for evaluation and feedback

2. Daily builds and rapid feedback from integration tests

3. A team and/or leader with the experience and instincts to make good decisions

4. A modular architecture that supports the ability to easily add new features

Principle 4: Defer Commitment

"In preparing for battle I have always found that plans are useless, but planning is indispensable."

Dwight Eisenhower

Emergency responders are trained to deal with challenging, unpredictable, and often dangerous situations. They are taught to assess a challenging situation and decide how long they can wait before they must make critical decisions. Having set a timebox for such a decision, they learn to wait until the end of the timebox before they commit to action, because that is when they will have the most information.

Principle 5: Deliver Fast

- Increase feedback cycle
- Respond to change quickly
- Stay ahead of competition

Principle 6: Respect People

1. **Entrepreneurial Leader**: People like to work on successful products, and highly successful products can usually be traced to excellent leaders. A company that respects its people develops good leaders and makes sure that teams have the kind of leadership that fosters engaged, thinking people and focuses their efforts on creating a great product.

2. **Expert Technical Workforce**: Any company that expects to maintain a competitive advantage in a specific area must develop and nurture technical expertise in that area. Companies that buy all the expertise they need will find that their competitors can buy it also. Companies that see no need for expertise will find that they have no sustainable competitive advantage. Wise companies make sure that appropriate technical expertise is nurtured and teams are staffed with the needed expertise to accomplish their goals.

3. **Responsibility-Based Planning and Control**: Respecting people means that teams are given general plans and reasonable goals and are trusted to self-organize to meet the goals. Respect means that instead of telling people what to do and how to do it, you develop a reflexive organization where people use their heads and figure this out for themselves.

Principle 7: Optimize the Whole

"A lean organization optimizes the whole value stream, from the time it receives an order to address a customer need until software is deployed and the need is addressed. If an organization focuses on optimizing something less than the entire value stream, we can just about guarantee that the overall value stream will suffer."

Implementing Lean

- Implementation of Lean is the whole organization exercise

- You must have buy-in from all stakeholders

- First choose goal of Lean implementation, you need it to track progress and know when baseline is there

- You are **never** done implementing Lean, it is ongoing process of relentless improvement

References

- Lean Software Development: An Agile Toolkit

  http://www.amazon.com/exec/obidos/ASIN/0321150783/poppendieckco-20

- Implementing Lean Software Development: From Concept to Cash

  http://www.amazon.com/exec/obidos/ASIN/0321437381/poppendieckco-20

- Leading Lean Software Development: Results Are not the Point

  http://www.amazon.com/exec/obidos/ASIN/0321620704/poppendieckco-20

- The Goal: A Process of Ongoing Improvement

  http://www.amazon.com/Goal-Process-Ongoing-Improvement/dp/088427
  1951/ref=sr_1_2?s=books&ie=UTF8&qid=1347376745&sr=1-2&keywords=t
  he+goal

- More books at http://www.poppendieck.com/reference.htm

09/12/12

@Dvyazelenko

vyazelenko.com