

# Lean Software Development Made Simple: How to Build Better Software with Less Waste and More Flow

Lean Beyond Manufacturing Series

Mohammed Hamed Ahmed Soliman

Published by [personal-lean.org](http://personal-lean.org), 2025.

Publisher: Personal Lean Publications  
Website: [www.personal-lean.org](http://www.personal-lean.org)  
For inquiries: [info@personal-lean.org](mailto:info@personal-lean.org)

No AI-generated text, translation or images were used in the production of this work.

All trademarks and names belong to their respective owners and are used for identification and educational purposes only.

While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

LEAN SOFTWARE DEVELOPMENT MADE SIMPLE:  
HOW TO BUILD BETTER SOFTWARE WITH LESS  
WASTE AND MORE FLOW

**First edition. August 6, 2025.**

Copyright © 2025 Mohammed Hamed Ahmed Soliman.

Written by Mohammed Hamed Ahmed Soliman.

# Table of Contents

Lean Software Development Made Simple   How to Build Better Software with Less Waste and More Flow.....	1
Chapter 1: Why Software Development Is Broken.....	8
Chapter 2: Lean Thinking for Developers.....	12
Chapter 3: Identifying Waste in Software Projects.....	17
Chapter 4: Build Quality In .....	28
Chapter 5: Fast Delivery and Flow .....	34
Chapter 6: Create Knowledge Through Experiments .....	40
Chapter 7: Delay Decisions Without Losing Speed .....	46
Chapter 8: Respect Your Developers Empowering Teams to Build Great Software.....	52
Chapter 9: Optimizing the Whole System.....	58
Chapter 10: Lean Metrics for Software Teams .....	64
Chapter 11: Lean Tools for Developers .....	70
Chapter 12: Real-World Case Studies .....	75
Bonus Section: Lean vs Agile vs DevOps .....	80
Final Words.....	84



# **Lean Software Development Made Simple**

**How to Build Better Software with Less Waste and More Flow**



# **C**ontents

## **Lean Software Development Made Simple**

### **Chapter 1: Why Software Development Is Broken**

### **Chapter 2: Lean Thinking for Developers**

- 1. Eliminate Waste**
- 2. Amplify Learning**
- 3. Decide as Late as Possible**
- 4. Deliver as Fast as Possible**
- 5. Empower the Team**
- 6. Build Quality In**
- 7. Optimize the Whole**

#### **Lean Is a Mindset, not a Method**

### **Chapter 3: Identifying Waste in Software Projects**

#### **What Is Waste in Software?**

#### **The 8 Types of Waste (DOWNTIME) in Software Development**

#### **How to Spot Waste Using Value Stream Mapping (VSM)**

#### **Real-World Insight: Waste Is Usually in the Wait**

#### **Take Action: How to Reduce Waste**

#### **Closing Thoughts**

#### **The 8 Wastes of Lean in a Software Context**

**How Value Stream Mapping Helps**

**Waste Is Contextual**

**Pro Tips for Identifying Waste**

**Key Takeaways**

## **Chapter 4: Build Quality In**

**Why Fixing Bugs Later Is Expensive**

**Test-Driven Development (TDD): Code with Confidence**

**Continuous Integration and Continuous Delivery (CI/CD)**

**Automation: Your Invisible Team Member**

**Clean Code & Refactoring Culture**

**Definition of Done = Quality Included**

**Quality Is Everyone's Job**

**Summary: Build Quality In**

**Action Steps**

## **Chapter 5: Fast Delivery and Flow**

**Why Fast Delivery Matters**

**Understanding Flow in Software Development**

**Optimize for Flow with Smaller Batch Sizes**

**Reduce WIP (Work In-Progress)**

**Lean Frameworks that Enhance Flow**

**Flow vs. Speed: Don't Sacrifice Value**

**Remove Bottlenecks**

**Measure Flow: Lead Time & Cycle Time**

**Summary Checklist**

**Key Takeaway**

## **Chapter 6: Create Knowledge Through Experiments**

**1. Learning Over Guessing**

**2. Prototyping: Fast, Cheap Learning**

**3. MVP: Minimum Viable Product**

**4. Feedback is Fuel**

**5. Lean UX: Designing for Learning**

**6. Measure What Matters**

**7. Culture of Experimentation**

**Summary**

**Key Takeaways**

## **Chapter 7: Delay Decisions Without Losing Speed**

**Why Delay Decisions?**

**YAGNI: "You Aren't Gonna Need It"**

**Feature Toggles: Deferring with Control**

**Designing for Change: Flexible Architecture**

**The Last Responsible Moment**

**How to Delay Without Drifting**

**Cultural Shifts Required**

**Summary**

## **Chapter 8: Respect Your Developers Empowering Teams to Build Great Software**

- 1. Why Respect Matters in Software Development**
- 2. Empowering vs Controlling**
- 3. Psychological Safety: The Foundation of Innovation**
- 4. Build a Culture of Trust**
- 5. Leadership for Technical Teams**
- 6. Reducing Cognitive Load**
- 7. Recognition and Growth**
- 8. Respect Goes Both Ways**

**Summary**

## **Chapter 9: Optimizing the Whole System**

**The Dangers of Local Optimization**

**End-to-End Thinking: From Idea to Production**

**Enter DevOps: Automating and Integrating the Flow**

**End-to-End Ownership**

**Feedback Loops as the System's Nervous System**

**Value Stream Mapping Revisited**

**System Optimization in Action: A Case Snapshot**

**Key Takeaways**

## **Chapter 10: Lean Metrics for Software Teams**

**Why Metrics Matter in Lean Software Development**

**The Danger of Vanity Metrics**

**Lean Metrics That Matter**

**Making Metrics Actionable**

**Metrics Without Context = Misleading**

**Visualizing Metrics: Dashboards and Boards**

**Principles for Using Metrics Wisely**

**Real-World Example**

**Key Takeaways**

## **Chapter 11: Lean Tools for Developers**

**Why Tools Matter in Lean Software Development**

**1. Kanban Boards: Visualizing Work and Limiting WIP**

**2. Value Stream Mapping (VSM): Seeing the Big Picture**

**4. 5 Whys: Root Cause Analysis Made Simple**

**5. Obeya Room (Digital or Physical War Room)**

**Bonus: Tech Tools Supporting Lean**

**Summary**

**Key Takeaway**

## **Chapter 12: Real-World Case Studies**

**Introduction**

**1. Case Study 1: A Startup's Lean Pivot**

**2. Case Study 2: Lean in a Large Enterprise**

### **3. Case Study 3: Lean Rescue of a Failing Government Project**

### **4. Common Patterns Across All Cases**

#### **Key Takeaways**

#### **Bonus Section: Lean vs Agile vs DevOps**

##### **Lean vs Agile vs DevOps: What's the Difference?**

##### **Tools Comparison Table**

##### **Cheat Sheet: Lean Software Development – Do's and Don'ts**

#### **Final Words**