Introduction to Lean Software Development

Practical Approaches for Applying Lean Principles to Software Development

Mary Poppendieck[1] and Tom Poppendieck[1]

Poppendieck.LLC, 7666 Carnelian Lane, Eden Prairie, MN, USA

Abstract. Long feedback loops are the biggest cause of waste in software development. They are the reason why well over 50% of all newly developed software is seldom or never used. Long feedback loops are the cause of seriously delayed projects, unmanageable software defect counts, and code bases that calcify because of their complexity.
Lean Software Development is all about shortening information feedback loops in the software development process and creating flow. The result is increased speed and quality along with lower cost. If this sounds unlikely, consider that in manufacturing, operations, and logistics, lean processes routinely deliver the highest speed, highest quality and lowest cost in extremely competitive environments. This tutorial will show you how to apply the principles that underlay lean manufacturing, lean logistics and lean product development to software development.

Format: The tutorial will be interactive. You will create a current value stream map of a real software development environment, learn how to apply lean tools to the environment, and then design a future value stream map.

Intended Audience: This program is designed for senior software development practitioners, team leads and managers who are considering lean software development for their organizations.

Learn how to:

1. Develop a value stream map for your software development organization and what to do once you have the map.
2. Assess the state of the basic disciplines which determine your software development process capability and organize a visual workplace so that everyone knows the most important thing to do next without being told.
3. Apply lessons from queuing theory to manage the software development pipeline.
4. Discover what's wrong with change approval and defect management systems, and how to reframe the development workflow to address both areas more effectively.
5. Use a financial model to avoid sum-optimization.

H. Baumeister et al. (Eds.): XP 2005, LNCS 3556, p. 280, 2005.
© Springer-Verlag Berlin Heidelberg 2005