



Lean Software Testing (Three Day Course)

Borrowing from lean manufacturing, agile development, and a great deal of personal experience, LSD provides methods for participants to improve the flow of software delivery by identifying waste, bottlenecks and obstacles. While the course does include deep dives into specific reference examples, the theme of is to allow graduates to customize existing methods. Starting with “what are we doing now?” students learn what to change next, how to design safe experiments, reduce risk, and continually improve.

Immediate Benefits of This Class

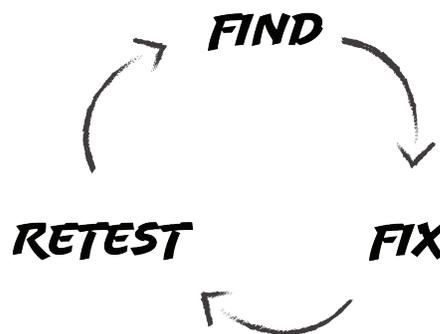
- 1 A solid understand of lean manufacturing and how that applies to software delivery.
- 2 A software testing method that plugs into a Traditional, Agile or even Continuous Delivery Software Model
- 3 Tools to measure performance of delivery.
- 4 Cycle time, work in progress, touch time, and takt time.
- 5 Understand how flow impacts throughput
- 6 Learn the Find -> Fix -> Retest loop, what it does to productivity ... and how to fix it.
- 7 Reduce time spent estimating and planning software projects.
- 8 Learn to communicate with tests as examples, and how to automate them.
- 9 Learn to communicate about status and priorities, inside and outside the team.
- 10 Tools to manage the time of an individual, a team, or multiple teams as a portfolio.
- 11 Map the value stream of the software, from concept to cash.
- 12 Learn story-mapping to generate a valid backlog of work quickly.

Day 1 - The Lean and the Delivery Cycle

We'll look at the flow of software, from concept to cash, including all of the roles on a project team and the effect of imaging work as a assembly-line. During this time we also examine the cost of work in progress, the definitions of lead time, and cycle time, and takt time.

We introduce the concept of a Kanban - to visualize our flow of work, with some examples, and introduce a simulation, which allows the team to experience software development at an accelerated pace. Participants observe delivery dynamics from a different perspective, then discuss what they have learned.

By the end of day one, students understand the challenges of delivery and test, especially for quick releases. They have have heuristics to find waste in tools, process, and documentation, and tools to measure and improve the flow of software through test.

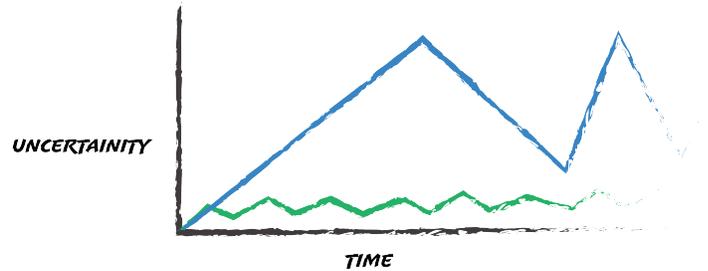


Day 2 - The Lean Organization

Once the students have a solid grasp of improvement concepts, we need to talk about what to improve.

The public version exposes the class to testing a variety of software and physical objects, asking questions about the role of testing and the promises testing can make. Students begin by learning quick attacks, then move to add analysis techniques, planning, communication, and negotiation. In the afternoon the class discusses test strategy, coverage, documentation, regression, as well as the tensions between improvisation and control, and how to manage that tension.

In private versions of the course we can work with the organization, using real software, artifacts, and work products as the examples, including actual testing, improving the test process, or re-imaging the test process as entirely different.



Day 3 - Making it Real

Day three integrates the idea of optimizing team performance (how) with organization and management performance (what/who). Under time pressure (the end of the course) and with a large amount of material, the instructors use a variety of techniques to get the class to agree on scope in order to get the best sessions.

In private training, the focus of day three is "what can the team do on Monday?", with a goal of creating a shared understanding of what experiments to conduct and what measures to start when the course ends. In public training the focus is similar, but more on individual value than the team as a whole.

Some of the possible modules for day three include

Acceptance Test-Driven Development	Managing Demand
Retrospectives for team improvement	Keeping up with pace
Continuous Delivery	Balancing Long Term Factors with Short Term Factors
Toggle Flags and DevOps	Automation Strategy
Writing Process Down - Wikis and Google Docs	Free and Cheap Tools
Lean Coffee, Affinity Mapping, and other Techniques	Quick and Easy Techniques
Whole Team Testing	

Class Format

The course is a combination of lecture, instructor-led, discovery, and conversation. After some initial static training we quickly work with students to figure out what the right module should be next. This allows them to understand delivery, and the project, from the view of the customer, who is trying to get as much done as possible in a limited set of time.

In private training we plan one to three discovery calls to get as much information as possible before the call, to customize the class to meet the needs of the client.