Agile Scrum Boot Camp

Length: 2 Days

COURSE CONTENT

Agile Thinking: In order for us to understand the benefits of Scrum and the nuances behind its framework, we begin with the history of agile methods and how relatively new thoughts in software development have brought us to Scrum.

1. How manufacturing has influenced software development
2. The origins of agile thinking
3. The Agile Manifesto
4. The complexity of projects
5. Theoretical vs. empirical processes overview
6. The "iron triangle" of project management.

The Scrum Framework: Here we'll ensure that we're all working from the same foundational concepts that make up the Scrum Framework.

1. The different Scrum roles
2. Chickens and pigs
3. Iterative development vs. waterfall
4. Self management concepts
5. Full disclosure and visibility
6. The Scrum framework overview

Implementation Considerations: Moving beyond Scrum's foundational concepts, we'll use this time to dig deeper into the reasons for pursuing Scrum. The key concepts of "empirical thinking" and "done" will be presented. We'll also use this time to begin a discussion of integrity in the marketplace and how this relates to software quality.

1. Traditional vs. Agile methods overview
2. Scrum: the silver bullet?
3. The Agile skeleton
4. A Scrum launch checklist

Scrum Roles: Who are the different players in the Scrum game? We'll review checklists of role expectations in preparation for further detail later in our session.

1. The Team Member
2. The Product Owner
3. The ScrumMaster

The Scrum Team Explored: Since the ScrumMaster is looking to protect the productivity of the team, we must investigate team behaviors so we can be prepared for the various behaviors exhibited by teams of different compositions. We'll also take a look at some Scrum Team variants.

1. The agile heart
2. Bruce Tuckman's team life cycle
3. Patrick Lencioni's Five Dysfunctions of a Team
4. Team ground rules
5. Getting Human Resources involved
6. The impact of project switching
7. The MetaScrum
8. The Scrum of Scrums
9. The importance of knowing when software is "done"
10. "Done" for multiple team integrations divided by function
11. "Done" for multiple team integrations divided by skill
12. "Done" for unsynchronized technologies
13. Internal outsourcing

Learnsoft
ENTERPRISE TRAINING

4350 Executive Drive, Suite 100

San Diego, CA 92121

San Diego (858) 546-1400

The Product Owner & Extracting Value: The driving force behind implementing Scrum is to obtain results, usually measured in terms of return on investment or value. How can we help ensure that we allow for project work to provide the best value for our customers and our organization? We'll take a look at different factors that impact our ability to maximize returns.

1. The priority guide
2. Product Backlog refactoring
3. Productivity drag factors
4. Fixed price/date contracts
5. Release management
6. Earned Value Management

The ScrumMaster Explored: It's easy to read about the role of the ScrumMaster and gain a better understanding of their responsibilities. The difficulty comes in the actual implementation. Being a ScrumMaster is a hard job, and we'll talk about the characteristics of a good ScrumMaster that go beyond a simple job description.

1. The ScrumMaster aura
2. Characteristics of a ScrumMaster candidate
3. The difficulties of being a ScrumMaster
4. A day in the life of a ScrumMaster
5. The importance of listening
6. Scrum's success depends on common sense

Meetings and Artifacts Reference Material: While most of this material was discussed in previous portions of class, more detailed documentation is included here for future reference.

1. A chart of Scrum meetings
2. The Product Backlog
3. Sprint Planning
4. The Sprint Backlog
5. The Sprint
6. The Daily Scrum
7. The Sprint Demo/Review
8. Why plan?
9. The Ideal Team Day
10. Scrum tools

Advanced Considerations: This section is reserved for reference material. Particular interests from the class may warrant discussion during our class time together.

1. Conflict management
2. Different types of Sprints
3. The ScrumMaster of the Scrum-of-Scrums
4. Metrics
5. Dispersed teams
6. Scaling
7. Developing architecture
8. Stage Gate/Milestone driven development
9. Inter- and Intra-project dependencies
10. Task boards, project boards
11. Scrum and CMM, "traditional" XP