Course Outline

Agile for Developers

Course Summary

Description

This course explores how adapting Agile values and principles will improve product development. It contrasts traditional SDLC methods with the most popular Agile methods to set the stage for benchmarking performance. Then it introduces a comprehensive set of Agile techniques and practices, and gives attendees the opportunity to simulate an end-to-end project while using them. Attendees discuss significant issues such as how to transition traditional roles of development within an Agile Framework.

The strength of this course comes from various exercises and discussions that give attendees hands-on experience with practical situations. The complete product life cycle is covered from inception to release planning and deployment, from managing a backlog to demonstrating results, and from prioritizing requirements to changing them. Attendees provide estimates using several techniques at successively lower levels of detail. They experience what it is like to participate in self-managing teams. Instructor-led discussions will debrief each exercise and apply learning to the attendee's environment. The class will also discuss the organizational, cultural and management implications of implementing Agile practices.

Objectives

By the end of the course, students will be able to:
- Produce work products and artefacts based on actual situations that demonstrate how to make the transition to an Agile environment

Topics

- Introduction – Fundamentals of Agility
- Value Driven Delivery – Identify Case Study and Team Members
- Stakeholder Engagement – Envision the Product
- Initiate an Agile Project – Planning Releases
- Agile Product Development Life Cycle planning
- Plan the Iteration (Part I)

- Plan the Iteration (Part II)
- Tools and Techniques for Managing Scrums
- Running the Sprint
- Sprint Review
- Sprint Retrospective
- Boost Team Performance
- Transitioning from Waterfall
- Wrap-up and Additional Information

Audience

The course is intended for Developers, Project Managers and other team members involved in the development of a product or service.

Prerequisite

None. However, it is recommended that participants have a basic understanding of a development product cycle, project management, business processes, business analysis or other IT functions.

Duration

Two Days

Agile for Developers

Course Outline

I. **Introduction – Fundamentals of Agility**
   A. Why Agile?
   B. Agile Manifesto, Agile Principles and Waterfall vs. Agile comparison
   C. Exercise 1a: Agile SCRUM Terms and Concepts Cheat Sheet
   D. The Scrum framework, roles, and Agile Product Life Cycle
   E. Exercise 1b: Challenges to building end-to-end systems with Agile
   F. Introducing Agile to the organization

II. **Value Driven Delivery – Identify Case Study and Team Members**
   A. Value-driven development: Understand why agile focuses so heavily on working products
   B. Agile Scrum characteristics
   C. Exercise 2a: Selecting the Case Study (selected by participants from current projects; basis for subsequent exercises)
   D. Assemble the Agile team
   E. Committed and Non-Committed members
   F. Product Owner responsibilities, characteristics and examples
   G. Exercise 2a: Select the "Product Owner"
   H. Building the Scrum team
   I. Scrum Master and committed team members
   J. Team collaboration
   K. Redefining traditional roles
   L. Exercise 2c: Agile PM and Developer (handouts; discussion)
   M. Exercise 2d: Build the Scrum Team (Scrum Master and Committed Members)
   N. Contrast with Waterfall using RACI matrix

III. **Stakeholder Engagement – Envision the Product**
   A. Understand the motivation behind stakeholder expectations and high-level functionality
   B. Business motivation model

C. Product envisioning
D. Evaluate functioning solutions for improvement opportunities
E. New product vision and scope. Breakdown epics into features and stories
F. High-level business and technical functionality
G. Exercise 3a: Product Vision Goals and Strategies
H. Agile coordination
I. Section 3b: Review Agile Checklist: Strategy, Release Planning, Sprint Planning, Agendas and Guidelines

IV. **Initiate an Agile Project – Planning Releases**
   A. Envision the product and project outcomes, common practices that work
   B. Exercise 4a: Adapting a Change-Driven Project Plan that Works

V. **Agile Product Development Life Cycle planning**
   A. Compile the Product Backlog
   B. Decision and acceptance criteria for user stories
   C. Planning Releases
   D. Prioritize Releases first, then Product Backlog items. Factors to consider
   E. Order Product Backlog items for initial Release Plan
   F. Exercise 4b: Create Release Plan
   G. Release Planning as an iterative planning process
   H. Coarse-Grain Estimating and Time-Boxing
   I. Estimate high-level items at sufficient detail for planning and prioritizing
   J. Exercise 5a: Elaborate Business Functionality (10 or more User Stories and associated technical functions)
   K. Estimate relative complexity
   L. Planning poker using story points

Course Outline (cont.)

M. Exercise 5b: Estimate Complexity (coarse-grain)
N. Team velocity
O. Establish the project time-box. Considerations: hours worked, resource availability …
P. Embrace the high-level Vision and Release Plan
Q. Exercise 5c: Establish Project Time-Box

VI. *Plan the Iteration (Part I)*
A. Sprint planning with the Product Owner
B. Contrast sequential and overlapping development
C. Create an accurate backlog at the Iteration level
D. Exercise 6b: Sprint "Zero" Activities
E. Anticipate Spikes
F. Create a Master Test Plan
G. Exercise 6c: Review Iteration Planning Checklist
H. Sprint planning meeting
I. Elements of successful Sprint planning
J. Prioritize user stories and identify PBIs for the Sprint
K. Exercise 6d: Confirm and Refine High-Priority Backlog Items
L. Exercise 6e: Post-Chapter Activity: Conduct a Daily Review and Retrospective

VII. *Plan the Iteration (Part II)*
A. Sprint planning with committed team
B. Plan and estimate Iteration tasks
C. User stories and associated tasks
D. Sprint planning, story size and task size
E. Produce task list
F. Estimate relative effort (fine-grain by team)
G. Planning poker using ideal days
H. Sprint backlog example
I. Exercise 7a: Identify and Estimate Sprint Backlog Tasks
J. Commit backlog items to the Sprint
K. Finalize the Sprint Plan

L. Exercise 7b: Review the Sprint Plan

VIII. *Tools and Techniques for Managing Scrums*
A. How to approach techniques in an Agile environment
B. Exercise 8a: Discussion – Project Activities for Scrum (communications, analysis and design, product quality, soft skills)
C. Sprint goal
D. Manage the Sprint backlog – key points
E. Information radiators and project status
F. Daily Scrum meeting
G. Scrum task board
H. Sprint Burn Down Chart
I. Product Burn Down Chart reflecting scope change
J. Exercise 8b: Create a Scrum Task board – Identify Work Streams

IX. *Running the Sprint*
A. Exercise 9a: Discuss Success Criteria for Self-Managed Teams
B. Self-managed teams
C. Paradigm shift in managing requirements
D. Team facilitation activities
E. Elaborate requirements details
F. Non-functional requirements
G. Select 'next priority task' and re-estimate task length (team member)
H. Challenges and opportunities in a global environment
I. Managing Scrums
J. Daily Scrum rules (committed v. non-committed speakers, handling issues)
K. Exercise 9b: Hold a Daily Scrum and Update Task Board
L. Authority to change the Sprint Backlog
M. How to manage requested changes during the Sprint

Course Outline

Agile for Developers

Course Outline