Writing Agile User Story and Acceptance Test Requirements Workshop

Course Summary

Description

Everyone complains that poor requirements are the major cause of project problems. Yet, like the weather, nobody does much about it, at least not effectively. Traditional approaches advocate writing voluminous requirements documents that too often don't seem to help much and may even contribute to difficulties. Agile goes to the opposite extreme, relying on brief requirements in the form of three-line user stories that fit on the front an index card and a few user story acceptance criteria that fit on the card's back. Surprise, as Mark Twain noted, in some ways it's even harder to write Agile's brief requirements effectively. This interactive workshop reveals reasons user stories and their acceptance tests can fall short of their hype, explains critical concepts needed for effectiveness, and uses a real case to provide participants guided practice writing and evaluating user stories and their acceptance criteria/tests.

Objectives

After taking this course, students will be able to understand:

- Major sources of poor requirements that cause defects, rework, and cost/time overruns.
- How Agile user stories and their acceptance criteria/tests address these issues.
- Difficulties that still afflict requirements in Agile projects and why they persist.
- Writing more effective user stories and acceptance criteria/tests.
- What else is necessary to produce working software that provides real value.

Topics

- Agile, User Story Fundamentals
- Requirements Are Requirements—or Maybe Not
- Writing More Suitable User Stories
- And User Story Acceptance Tests

Audience

This course has been designed for product owners, analysts, developers, and other Agile (and other) project team members who are or should be involved in defining requirements.

Prerequisites

There are no prerequisites for this course.

Duration

One day

ProTech "Charting the Course ...
protechtraining.com
... to Your Success!"

Writing Agile User Story and Acceptance Test Requirements Workshop

Course Outline

I. Agile, User Story Fundamentals
- A. Agile Manifesto's relevant points
- B. Characterization of traditional approaches
- C. Waterfall and big up-front requirements
- D. Agile's sprints and backlogs alternative
- E. Agile project team roles
- F. User story "As a <role>…" (Card)
- G. User story acceptance criteria (Confirmation)
- H. Estimating user story size
- I. Splitting and refining
- J. Prioritizing and allocating to backlogs/sprint
- K. Constructing/implementing (Conversations)
- L. Reviewing, retrospectives
- M. Grooming backlog and reprioritizing
- N. Write Needed User Stories
- O. Define their Acceptance Criteria
- P. Review Your User Stories/Criteria

II. Requirements Are Requirements—or Maybe Not
- A. User stories are backlog items, features
- B. Chicken and egg relation to use cases
- C. Issues and inconsistencies
- D. Business vs. product/system requirements
- E. "Levels Model" of requirements
- F. Other mistaken presumptions
- G. Requirements overview
- H. Where user stories should fit, do fit instead
- I. Conversation conundrum

III. Writing More Suitable User Stories
- A. Problem Pyramid tool to get on track
- B. Using the Problem Pyramid
- C. Business Requirement User Stories
- D. Issues identifying requirements
- E. Product owner and business analyst roles
- F. Project team participation
- G. Dictating vs. discovering
- H. Data gathering and analysis
- I. Planning an effective interview
- J. Controlling with suitable questions
- K. Then a miracle occurs…

IV. And User Story Acceptance Tests
- A. Missed and unclear criteria
- B. Turning criteria into tests, issues
- C. How many tests are really needed
- D. Test design techniques
- E. Checklists and guidelines
- F. Decision trees, decision tables
- G. Boundary testing
- H. Testing is main means to control risk
- I. Defects and new user stories
- J. Testing that user story focus misses
- K. Reactive vs. proactive risk analysis
- L. Given, when, then format
- M. Write User Story Acceptance Criteria
- N. Design their Tests
- O. Review Your User Stories/Tests