Certified Scrum Developer

Course Summary

Description

This five day course will help you to become an effective Developer in a Scrum environment. The highly sought after training contributes towards the requirements of the Scrum Alliance's Certified Scrum Developer program.

Objectives
At the end of this course, students will have learned Agile Engineering Practices, including:

- Study of architecture and design.
- An in-depth look at the way Agile teams work together.
- Study of Test Driven Development (TDD).
- An introduction to the practice of Refactoring.
- An introduction to the key practices of Continuous Integration (CI).
- Fulfill the requirements of the Training

Topics

- Welcome and Introduction - Scrum basics
- Story Mapping and Story Writing
- Acceptance Criteria
- Getting Started with Technical Practices
- Database Strategies
- Graduation

Audience

This course is ideal for those who desire to create high-performance product development teams. Professional developers will gain tremendous understanding about Scrum's amazing transformational power and the critical role of the Scrum developer.

Prerequisites

There are no prerequisites for this course.

Duration

Five days

BERTEIG WORLDMINDWARE

ProTech "Charting the Course ...

protechtraining.com

... to Your Success!"

Certified Scrum Developer

Course Outline

I. Welcome and Introduction - Scrum basics
- A. Five Day Course Outline
- B. Discussion: Introduction to the class
- C. Discussion: What is scrum and where does the developer fit into this?
- D. Exercise: Build your own Scrum

II. Story Mapping and Story Writing
- A. Exercise: Personas
- B. Discussion: Story mapping
- C. Exercise: Story writing

III. Acceptance Criteria
- A. Exercise: Acceptance criteria
- B. Discussion: Introducing the Gherkin format
- C. Exercise: Write some Gherkin
- D. Exercise: Learn the importance of correct language
- E. Discussion: Estimates

IV. Getting Started with Technical Practices
- A. Exercise: Pair Draw [10 min]
- B. Introduction to Pair Programming [10 min]
- C. Exercise: FizzBuzz in Pairs [50 min]
- D. Exercise: Add tests after the fact to FizzBuzz
- E. Exercise: Simple Design and TDD in LEGO® [30 min]
- F. Exercise: Simple Design and TDD in Code [60 min]
  1. Simple design and TDD continued
  2. Discussion: Introduction to Mocking and Stubbing [5 min]
  3. Exercise: Mocking and Stubbing in code [30 min]
- G. Emergent Design
  1. Exercise: Mocking / Stubbing / TDD continued
  2. Exercise: Technical Debt in LEGO®
  3. Exercise: Technical Debt in code
  4. Discussion: Architectural debt
  5. Discussion: Strategies for reducing debt
- H. Agile Testing
  1. Discussion: Agile Testing
  2. Revisit Pair Programming
  3. Refactoring with existing tests
  4. Refactoring without tests
  5. Clean Code
  6. Discussion: Keeping the code clean
  7. Exercise: Cleaning code
- I. Continuous Integration (CI)
  1. Discussion: Continuous Integration Overview
- J. Continuous Delivery
- K. Discussion: Continuous Delivery
- L. Exercise: Build your own CI/CD pipeline

V. Database Strategies
- A. Acceptance Test Driven Development (ATDD)
- B. Discussion: Purpose and workflow of ATDD
- C. Exercise: Automate with cucumber
- D. Mobbing On Code
- E. Recap, Retrospective and Q&A

VI. Graduation
- A. What Did You Learn?
- B. Certification
- C. Feedback Forms