Professional Scrum Developer

Duration: *2 Day(s)*

Course Overview

Professional Scrum Developer™ (PSD) is a 3-day hands-on course where you will experience how to deliver quality software using Scrum with Agile and DevOps practices. Students will work as a Scrum Team or multiple Scrum Teams depending on class size, creating real code on a realistic software system.

Review this course online at https://www.alta3.com/courses/SCRUM_30PSD

Objectives

- Understand the Scrum framework in detail
- Minimize project uncertainty and risk by applying Agile principles
- Practice Scrum techniques through actual application in the classroom
- Learn to measure and evaluate software development progress effectively

Who Should Attend

- Software Developers
- Scrum Team Members
- Technical Leads
- Agile Practitioners

Prerequisites

The Professional Scrum™ Developer course is suitable for any member of a software development team, including architects, programmers, database developers, testers, others with some technical knowledge. It is recommended that participants have a basic understanding of project management and business processes and business analysis. Participants will be expected to create and modify code.

Course Outline

Section 1: Scrum Fundamentals

1. The case for agile and Scrum
2. Empirical model
3. Defining Scrum
4. Scrum roles
5. Scrum artifacts
6. Scrum activities
7. Complexity and Scrum
8. Waterfall and Scrum

Section 2: Refining the Backlog

9. Evolution of requirements

10. Agile estimation
11. Levels of planning
12. Acceptance Criteria
13. Stories, Epics, Themes
14. Release Planning

Section 3: Application Life Cycle Management

15. ALM overview
16. Tracking sprints and the backlog
17. Branching strategies
18. Continuous Integration and deployment
19. Release models
20. Tools

Section 4: Quality and Scrum

21. Quality management
22. Technical Debt
23. Pair programming
24. Definition of done
25. Definition of ready
26. Testing automation/naming/coverage
27. BDD and TDD cycles
28. Test management and naming
29. Explorative testing
30. Specification by example

Section 5: Emergent Architecture

31. Architectural patterns
32. Emergence
33. Patterns and anti patterns

Section 6: Challenges with Quality and Scrum

34. Refactoring patterns
35. Clean code
36. SOLID code
37. Scrum challenges
38. Adoption patterns

Section 7: Additional notes and questions