



Applying Professional Scrum For Software Development

Duration: 3 days; Instructor-led

WHAT YOU WILL LEARN

The 3-day Applying Professional Scrum for Software Development course provides a real-world view of what it is like to build software with Scrum. Throughout the course, students collaborate as a team in a series of Sprints where they apply modern engineering practices and use the Scrum framework to cope with changes. Students experience how to deliver quality software using Scrum with Agile and DevOps practices. Students learn how to develop increments of potentially releasable functionality from a realistic Product Backlog over the three days. Students concurrently do requirements engineering, design, development, Testing, Integration, and deployment within a single iteration. The course teaches how Agile engineering practices and supportive DevOps practices improve a team's capabilities even more.

AUDIENCE

The Applying Professional Scrum for Software Development course is intended for all Scrum Team members to develop and deliver software-based systems. It is including architects, analysts, programmers, database developers, testers, managers, IT-Operations, and others, including Scrum Masters and Product Owners. This course provides the greatest value if the whole Scrum Team (Developers, Scrum Master, and Product Owner) attend together and experience the power of real teamwork. Teams that attend together are not only able to apply learnings to their work immediately but are also able to recall their classroom experiences to overcome particularly challenging times.

Of course, this class is also valuable to individuals attending without their entire team. These students will experience working on a Scrum Team during the class. They will be able to transfer their learnings to their teammates, often convincing their teammates of the advantages of using Scrum, Agile, and DevOps practices.

PREREQUISITES

There is no prerequisite, but attendees will be able to make the most of the class if they:

- Have studied the Scrum Guide (<http://www.scrumguides.org>)
- Have taken the Scrum Open (free online assessment)
- Have experience in software development either as a developer, tester, UI designer, Business Analyst, DBA, or DevOps engineer.

COURSE OBJECTIVES

- Experience real collaboration between Developers, Product Owner, and Scrum
- Master in the quest of building and delivering a high quality and valuable product.
- Build and deliver working software by applying modern Agile engineering practices and supportive DevOps tools.
- Understand the synergy between the various elements of Scrum and complementary practices.

OUTLINES

Module 1: Agile And Scrum Framework

- What is time-boxing, and why time-boxing?
- What is the Empirical Process?
- Scrum Framework – in Brief

Module 2: Initiation Of Development Using Scrum

- Introduction to a class project
- Scrum Roles
- Definition of Done

Module 3: Product Backlog To Bring Transparency

- Prepare product backlog
- Refine product backlog
- Ordering product backlog

Module 4: Agile Estimation & Planning

- Why estimate and how to estimate?
- Relative Estimation & preparing Release plan
- Estimating Product Backlog

Module 5: Sprint Simulation Using DevOps Tools

- Sprint simulations
- Agile Metrics & Reporting
- Team dynamics and dysfunction

Module 6: Specification By Examples Workshop

- Challenges in Collaboration between Business and Development Team
- How to write a better specification
- Why Examples needed for specification
- Writing Examples in Gherkin

Module 7: ATDD And BDD Workshop

- Principles of Acceptance Test-Driven Development
- Consequences of ATDD on the Sprint Planning and Sprint Review
- Practicing BDD with Cucumber/Behave/SpecFlow

Module 8: Test-Driven Development

- Test-Driven Development (TDD) approach and its benefits
- Improving the quality and design of the code by TDD
- Different types of tests: unit, Integration, functional

Module 9: Unit Testing With Unit Testing Framework

- Understand the life cycle of a test in the Unit Testing framework
- Respect for the principle of isolation and control of test results
- Use of simulation frameworks

Module 10: Software Design

- Definition and principles of frameworks
- Emergent Design
- Principle of scalability and implementation

Module 11: Refactoring

- The consequence of technical debt and the risk of "breaking" the code
- Three stages of refactoring
- Support of TDD and ATDD during refactoring

Module 12: DevOps

- Why what and how DevOps, Infrastructure as code
- 3-ways of DevOps, Culture, Automation, Measure and Share (CAMS)
- Continuous Integration, delivery, and deployment

Module 13: Agile Testing

- Importance of Agile Testing
- Ensure better quality product sprint by sprint
- Test Automation Frameworks
- Test quadrants and test pyramids