

Agile Project Management with Scrum - 3 Days

Techniques for Accelerated Software Development

Course 918 Overview

- You Will Learn How To**
- Develop higher quality complex software systems and products in reduced time using Agile practices
 - Mitigate uncertainty and risk by applying Agile values and principles
 - Apply the Scrum framework to meet the specific operational needs of your business
 - Focus on continuous improvement through embedded learning in the workplace
 - Adapt techniques to apply Scrum to large-scale and geographically-distributed projects
 - Incrementally change the culture of the wider business to benefit from agile methods
- Course Benefits** Traditional software development can be a cumbersome process, dominated by uncertainty and risk. Agile methods and the Scrum framework address this problem by promoting open collaboration and adaptability throughout the project life cycle. These methods focus on developing value-added software in short amounts of time. In this course, you learn how to apply Agile and Scrum techniques to manage software development projects.
- Who Should Attend** Project and programme managers, software architects, systems analysts, team leaders, developers and anyone interested in applying Scrum and Agile methodologies.
- RealityPlus™** Throughout this course, case studies and experiential activities immerse you in an authentic Agile project management environment. Activities include:
- Identifying candidates for the Product Owner role
 - Simulating a three-iteration project from start to finish
 - Estimating product feature size
 - Choosing features to implement in an iteration
 - Planning a distributed, multiteam project
 - Interpreting quantitative data to track a project's progress
 - Accelerating learning with tools and techniques for retrospection
 - Recommended solutions for everyday challenges faced by ScrumMasters

Agile Project Management with Scrum - 3 Days

Techniques for Accelerated Software Development

Course 918 Outline

Introduction to Agile Project Management

- Articulating Agile values and principles
- Comparing Agile with traditional, masterplan methods
- The myths and realities of Agile software development
- Locating Scrum in empirical process theory

Fundamentals of Scrum for Dealing with Uncertainty and Risk

Identifying the roles and their responsibilities

- Prioritising requirements and driving the project through the Product Owner
- Establishing the ScrumMaster as a fundamentally new management role
- Shaping the self-managed Feature Team
- Relating to external stakeholders

Managing product delivery

- Creating a product backlog as a list of requirements and technical issues
- Iterating development through cycles of one month or less
- Time boxing to establish a regular rhythm of energized work

Tools for tracking and monitoring a project

- Capturing requirements as user stories
- Establishing performance targets using a team's velocity
- Plotting remaining work with burn down charts and parking lot diagrams
- Managing development tasks with task boards

Planning an Agile Project

Focusing on business value

- Delivering business-valued functionality with Release Planning and Sprint Planning
- Collaborating with customers to manage risk and uncertainty
- Demonstrating implemented features as "potentially shippable increments"

Identifying features for development in an iteration

- Breaking large requirements into testable, estimatable features
- Concretising a Sprint Goal through feature selection
- Identifying development tasks in the Sprint Backlog

Fostering Self-Management within the Development Team

Creating the optimal working environment

- Staffing the Feature Team
- Protecting the team from outside interference
- Making project progress visible and open

Transitioning to self-management

- Facilitating cross-functionality and team learning
- Empowering the team to control their own development process
- Adapting management roles external to the development effort

Running iterations

- Microplanning through Daily Scrum meetings
- Producing a quality-assured, business-valued product
- Determining what it means to be "done"

Managing Change

Generating rapid feedback through Sprint Reviews

- Demonstrating completed functionality to the customer
- Fostering collaboration with stakeholders through discussion of "done" increments
- Revising estimates on the basis of experience

Reviewing iterations through Sprint Retrospectives

- Reflecting on lessons learned
- Embedding new knowledge to maintain continuous improvement

Applying Agile throughout Your Organisation

Scaling for large projects

- Working with large Product Backlogs
- Scaling the Product Owner role
- Coordinating Feature Teams and Component Teams

Managing distributed development

- Release and Sprint Planning for distributed teams
- Ensuring effective communication across time zones
- Holding distributed Sprint Reviews and Sprint Retrospectives

Establishing momentum for successful adoption of Agile methods

- Assessing the organisation's readiness for Agile adoption
- Creating an Improvement Backlog for ongoing improvement