

## Applying Scrum - Agile Principles and Practices

Agile is an approach to software development that has rapidly become mainstream. Incorporating the Agile philosophy into IT organizations has been shown to more rapidly deliver working software, increase quality, and decrease time-to-market. Using a mixture of techniques that together form modern Agile practices such as Scrum, Extreme Programming (XP), user stories, test-driven development, etc, teams can significantly improve their productivity and morale.

This course explains this approach's values, key techniques, and how each Agile team member contributes to success. In this course you will learn what makes Agile iterative, the fundamentals of the Agile approach, Scrum basics, backlogs and user stories, and Agile project and release planning. You will also learn how to plan, run, test, and close a sprint, which is the basic time box for an Agile project.

### Objectives:

- Understand key Agile and Scrum concepts, terms, and practices
- Be able to explain Agile and Scrum benefits and key techniques
- Become familiar with the progression of key Scrum activities and artifacts
- Prepare to participate in an upcoming Agile project

### Audience:

This course is intended for all who will actively participate in an Agile project, including Scrum team members, ScrumMasters, Product Owners, and Subject Matter Experts. People on the edges of the project (secondary customers, management staff, and participants in related processes) will also benefit.

### Pre-requisites:

None

### Duration:

1 day

## Outline:

### 1. Agile Fundamentals

- Values and history of the Agile Manifesto
- Agile vs. Traditional Approaches
- A blending of Agile methods including Scrum, XP, Unified Process, etc.

### 2. Scrum Overview

- Scrum history and process flow
- Scrum Roles and responsibilities
  - 1.) Product Owner
  - 2.) Scrum Master
  - 3.) Project Team

### 3. Agile Requirements

- Key principles for Agile requirements
- User Stories – includes user story benefits, format, and examples

### 4. Estimating and Release Planning

- Planning Overview
- Why do we plan?
- What makes a good plan?
- How is Agile planning different?
- Creating the product backlog
- Setting priorities – how the product backlog gets prioritized
- Planning releases – estimating the product backlog, preparing for Sprint planning

### 5. Planning a Sprint

- Sprint backlog –definition, how it is established, tips and trick for creating and managing it
- Conducting the Sprint planning meetings
- Part 1: Selecting a subset of the product backlog, how to determine the sprint scope including carryover

- Part 2: Developing the sprint backlog, how to avoid carryover, estimate velocity, and establish a sprint goal

### 6. Running a Sprint

- Understanding a Sprint's rhythm
- Conducting the daily Scrum – the focus and conduct of the daily Scrum
- What's included and what's excluded, the value of the daily Scrum
- Managing sprint scope – tips and techniques for insulating the team from distractions and removing impediments; how to handle the sprint being behind or ahead
- Warning signs and how to address them – covers common pitfalls and avoidance strategies and recipes for failure
- Tracking progress/burndown, methods for managing task completion
- Developing sprint tasks – includes task granularity, dependencies, scheduling and assignment

### 7. Closing a Sprint

- Conducting a sprint review – what is expected and covered in a sprint review
- Conducting a sprint retrospective – purpose and scope of a retrospective
- Inspecting and adapting the process
- Closing the project – summarization and definition of done