

**Agile XP**

**Course Number:** AGL-192  
**Duration:** 4 days

**Overview**

This Agile XP training course teaches attendees methodologies that combine Agile techniques with Extreme Programming (XP) principles, focusing on development speed, code quality, and team collaboration. The Agile XP framework contains 12 practices divided into four areas derived from the best software engineering practices. Participants learn how to put these Agile XP strategies into practice immediately.

**Prerequisites**

No prior experience is presumed.

**Materials**

All Agile XP training attendees receive comprehensive courseware.

**Software Needed on Each Student PC**

For in-person deliveries, attendees do not need computers for this course. We will provide full classroom setup instructions that will include seating in small groups, with supplies such as flipcharts, sticky notes, markers, and pens for the attendees and a projector and Internet connection for the instructor's laptop.

Online deliveries for this interactive training will use an online meeting platform (such as Zoom, WebEx, GoTo, or Teams) to have face-to-face contact online, including use of breakout rooms for group activities.

**Objectives**

* Understand the Agile and Scrum Framework
* Understand coding standards
* Work with Scrum roles, artifacts, and events
* Write user stories
* Understand the Principles of Behavior-Driven Development (BDD)
* Improve the quality and design of the code by testing
* Incorporate best practices and understand the limitations
* Work with Continuous Improvement (CI)

**Outline**

* Agile Overview
  + Manifesto & Principles
  + What is Extreme Programming (XP)
  + How and why XP with Scrum
  + XP Values and Rules – 12 practices of XP overview
* Agile and Scrum Framework
  + What is time-boxing and why time-boxing?
  + What is Empirical Process?
  + Scrum Framework – in Brief
* Scrum with XP Teams
  + Roles and Responsibilities
  + Creating Shared understanding
  + Coding Standards
  + Collective Code Ownership
  + Simple Design
  + System Metaphor
  + Whole Team
* Scrum & XP Overview
  + Scrum Roles, Artifacts, and Events
  + Definition of Done
  + Development Team Practices
  + Whole Team
* Scum & XP Backlog Planning
  + User Story and Estimation
  + User Story Writing Workshop
  + Acceptance Criteria
  + Splitting User Story
  + Running Spikes on Complex stories
  + Release Planning
  + Planning Game
  + Small Releases
  + Metaphor
  + Iteration Planning
  + Refine Product Backlog: Prioritization, Sequencing MVP & MMF
* BDD workshop – Collective Ownership and Testing (optional-extra day)
  + Principles of Behavior-Driven Development (BDD)
  + Why are Examples needed for specification?
  + Writing Examples in Gherkin
  + Practicing BDD
* Test Driven Development
  + Test Driven Development
  + Pair Programming
  + Improving the quality and design of the code by testing?
  + Different types of tests: unit, integration, functional
* Refactoring
  + The consequences of technical debt and the risk of “breaking” the code
  + Where? When? How refactoring?
  + Support of TDD and ATDD during refactoring
  + Best practices and limitations
  + The various categories of refactoring tools
* Introduction to Continuous Integration
  + Continuous Integration Overview
  + Continuous Improvement
  + Refactoring
  + CI and DevOps
  + Small Releases
* Conclusion