

**Agile Engineering and Quality**

**Course Number:** AGL-182  
**Duration:** 5 days

**Overview**

This live online or in-person Agile Engineering and Quality training teaches Agile teams how to implement Agile technical practices from methods such as Scrum, eXtreme Programming (XP), Feature Driven Development (FDD), OO, and Reactive and Functional Programming. Attendees develop new team engineering norms they can immediately put into practice.

**Prerequisites**

Participants must have strong knowledge of Java and foundational knowledge of Agile.

**Materials**

All 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**

* Learn to code the agile way
* Increase predictability and time to market with less stress on your team
* Practice agile by doing agile
* Understand and codify quality and engineering excellence
* Understand the purpose of Object-Oriented, Functional, and Reactive Programming
* Discover key techniques for increasing your team’s effectiveness
* Tune and customize agile in your organization
* Understand where and when to design as you iterate
* Incorporate transparent and accurate communication with customers and leadership
* Create mocks
* Understand the intent of microservices and why design/architecture are critical to success in Agile
* Increase predictability and deliver more consistently
* Remove impediments and improve performance
* Create a culture of pride and software development expertise

**Outline**

* Introduction
  + Survey of the Scrum Guide
  + Agile Manifesto Principles
  + I See Agile
* Sprint 0: Release Planning
  + Agile Framework
  + Agile Advantages and Potential
  + The Agile Manifesto and Principles
  + Lean Practices
  + eXtreme Programming
* Sprint 1: Sprint Planning & TDD
  + Ready?
  + Agile Estimation
* Sprint 2: Quality Code
  + Engineering Excellence
  + Code Quality
    - Written Metrics
  + Lightweight Design
* Sprint 3: Mocks
  + Mocks
  + Fakes, Stubs
  + Mockito
  + LAB: Mock Objects
* Sprint 4: Microservices
  + Microservices
  + RESTful
  + Testing Services
* Sprint 5: Reactive Programming
  + Functional Programming
  + Reactive Programming
* Sprint 6: DevOps
  + The DevOps Pipeline
  + Sonar Lint
  + Jenkins
* Sprint 7: Enhancing Performance
  + Sprint Review
  + Sprint Retrospective
    - Innovation and Learning Experiments
  + Keeping Score – Task Boards and more
* Sprint 8: BDD (Optional)
  + Acceptance Criteria
  + Introduction to Cucumber
  + Writing scenarios with Gherkin
  + Setup and Tear Down
* Conclusion
  + The Self Managing Team
  + Lean/Agile Center of Enablement
* Conclusion