

**OO Design and Modeling in Agile**

**Course Number:** AGL-122
**Duration:** 5 days

**Overview**

Accelebrate's Agile Training: Object Oriented Design and Modeling in Agile training teaches students how to create UML diagrams, including conceptual level diagrams, sequence diagrams, and use case diagrams.

**Prerequisites**

Students should have some knowledge of Agile/Scrum. Prior OO and UML experience would be helpful.

**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**

All students will start with a rough set of high level stories. During the course they will learn to deconstruct the stories and continually groom the product backlog according to the Product Owner’s direction (instructor).

**Outline**

* Introduction
* Applied OO
	+ Definition and Motivation for OOP
	+ Ensure Understanding of OOP Fundamentals
	+ OOP "First Principles"
* UML Essentials
	+ Use Cases
	+ Class Diagrams
	+ Sequence Diagrams
	+ "Turning Straw Into Gold" Using UML
* Agile Development
	+ Overview of agile software development
	+ Agile Manifesto Values and Principles
	+ Introduction to iterative coding practices
* Release Planning
	+ First Time Estimates
	+ Sizing Your Stories
	+ Velocity-Driven Release Planning
	+ Applying Themes
	+ Leveraging Use Cases For "Check and Balance"
* Product Owner Roles & Responsibilities
	+ Will the Real Product Owner Please Stand Up
	+ Product Backlog & Requirements
	+ Roles and Personas
	+ Story Workshops
	+ Spikes and Special Stories
	+ Identify and Define User Roles &: User Story Writing
* Grooming The Backlog
	+ Prioritization
	+ Looking Ahead
	+ Groom Continually
	+ Value Assessment
	+ Risk-Based Prioritization
	+ Running a Successful Sprint Planning Meeting
* Commonality and Variance
	+ Techniques for Translating From Stories and/or Use Cases to a Class Diagram
	+ Fundamentals of Commonality/Variability Analysis (CVA)
	+ CRC Cards
	+ How to Handle Variations as We Get New Requirements
	+ Understanding and Using Factories
* Delegation
	+ Delegation and Why it is so Powerful
	+ How Various Design Patterns Leverage Delegation
	+ Adapter Pattern
	+ Strategy Pattern
* Refactoring
	+ What is Refactoring
	+ Why Refactor
	+ Identifying Code Smells
	+ How the IDE can assist in refactoring
* Using Abstraction
	+ Understanding the Template Method Pattern
	+ Importance of Depending on Abstraction
* Model-View-Controller
	+ MVC Principles and the Motivation for Using MVC
	+ Differentiating between Model 1 & Model 2 Architectures
	+ Benefits of the Front Controller Pattern
	+ Using the Observer Pattern to reduce coupling
* Managing Access
	+ The Proxy Pattern and the Motivation for it
	+ How a Dynamic Proxy Works
	+ Why the Dynamic Proxy Offers a More Flexible Proxy Solution
* Dynamic Responsibilities
	+ Using the Decorator Pattern to Bring Flexibility to Designs
	+ Adding Functionality and Flexibility
	+ Decorator Pattern - Class Diagram
	+ Decorators in the Java/IO Pattern
	+ Designing a Custom I/O Decorator
* Conclusion