

AI-311: Using Coding Agents, Github spec-kit and Agentic skills for Spec-Driven and Test-Driven Development (SDD and TDD)

Course Length: 1 day or 2 half days

Course Description:

AI-powered coding agents such as Claude Code, OpenAI Codex, GitHub Copilot, Google Antigravity or Amazon Kiro are rapidly transforming how software is designed, implemented, and maintained. This course teaches participants how to move beyond simple agentic coding or “vibe-coding” and adopt a systematic software development framework, based on the Spec-Driven Development (SDD) and Test-Driven Development (TDD) methodologies, that improve productivity without sacrificing quality, governance, or maintainability.

Participants will learn how to use the open-source Github `spec-kit` and a popular agentic skill to implement the Spec-Driven and Test-Driven Development (SDD and TDD) methodologies respectively. SDD’s structured approach spans the whole software development life cycle (SDLC) and ensures that software development remains aligned with business intent, technical constraints, and long-term sustainability. The complementary TDD approach, by enforcing the Red-Green-Refactor cycle before writing any production code, makes sure that the generated code faithfully implements the specified requirements and nothing more.

The course combines conceptual foundations with extensive hands-on exercises done on a popular large open-source code base, enabling participants to immediately apply these techniques to real-world projects.

Training objectives:

By the end of this training, participants will be able to:

- Understand the limitations of “vibe-coding” and the need for a more structured methodology which spans the whole software development life cycle (SDLC)
- Understand the main principles, advantages and disadvantages of Spec-driven and Test-driven Development (SDD and TDD)
- Learn about the installation and basic configuration of `spec-kit` and the TDD agent skill
- Explain the meaning and importance of SDD and TDD steps
- Learn about advanced features and security aspects of SDD and TDD

Main topics:

- Limitations of “vibe coding” and the need for a more structured approach which spans the whole software development life cycle (SDLC)
- Main concepts of Spec-driven and Test-driven Development (SDD and TDD)
- Installation and basic configuration of `spec-kit` and the TDD agent skill
- SDD workflow:
 - Constitution enforcing TDD
 - Specification
 - Planning
 - Task breakdown
 - Implementation
- Advanced Features and Security

Structure: roughly 40% lecture, 60% hands on lab exercises.

Target audience: Software developers and testers as well as their technical managers who want to understand and use the Spec-driven and Test-driven Development (SDD and TDD) methodologies with a coding agent, Github `spec-kit` and a TDD agent skill.

Prerequisites:

- Experience with Claude Code, OpenAI Codex, Github Copilot, Google Antigravity or Amazon Kiro
- Experience using ChatGPT or similar chat-based AI tools
- General programming skills, experience with at least one object-oriented programming language

This training is part of the Agentic AI portfolio of Component Soft which explores essential topics, such as:

- AI-101: Intro to GenAI with Large Language Model (LLMs) and LLM-based apps.
- AI-241: Using Github Copilot for Agentic Coding
- AI-271: Using Claude Code for Agentic Coding
- AI-281: Using OpenAI Codex for Agentic Coding
- AI-311: Using Coding Agents and spec-kit for Spec-Driven and Test-Driven Development (SDD and TDD)
- AI-261: Using Amazon Kiro (I) for Agentic Coding
- AI-361: Using Amazon Kiro (II) for Spec-Driven and Test-Driven Development (SDD and TDD)
- AI-453: Agentic AI Application Development with LLMs
- AI-454: Agentic AI Application Development with LLMs (extended version)