AI-242: Using Github Copilot and spec-kit for agentic coding and spec-driven development (SDD)

Course Length: 2 days or 4 half days

Course Description:

AI-powered coding tools such as GitHub Copilot are rapidly transforming how software is designed, implemented, and maintained. This course teaches participants how to move beyond simple prompt-based coding and adopt a systematic, agentic approach that improves productivity without sacrificing quality, governance, or maintainability.

Participants will learn how to use Copilot's advanced chat and agent modes without or with spec-kit to implement "vibe coding" and Specification-Driven Development (SDD) respectively. SDD's structured methodology 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 course combines conceptual foundations with extensive hands-on exercises, 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 differences between the various modes in Copilot Chat
- Understand the concepts of MCP servers and tools, and configure and use them in Copilot
- Create new applications and modify existing ones consisting of multiple files
- Use Copilot to create and run unit and integration tests
- Generate documentation for applications based on source code and user prompts
- Use Copilot for source control tasks

- 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 Specification-Driven Development (SDD)
- Explain the meaning and importance of SDD steps
- Use SDD in greenfield projects
- Use SDD in brownfield projects

Main topics:

GitHub Copilot and Agentic Coding

- Copilot Chat modes: Ask, Edit, Plan, and Agent
- Configuring and using MCP tools and toolsets
- Using Copilot to create and modify multi-file applications
- Using Copilot to create and run unit and integration tests
- Using Copilot for documentation
- Using Copilot for source control
- Configuring and using asynchronous background and cloud-based coding agents

Specification-Driven Development (SDD)

- Limitations of "vibe coding" and the need for a more structured methodology which spans the whole software development life cycle (SDLC)
- What is Specification-Driven Development (SDD)?
- Installation and basic configuration of spec-kit
- SDD workflow:
  - Constitution
  - Specification
  - Planning
  - Task breakdown
  - Implementation
- Applying SDD in greenfield projects
- Applying SDD in brownfield projects

Structure: roughly 50% lecture, 50% hands on lab exercises. The lab exercises are executables with Python, JavaScript/Typescript, Java, C# or C/C++.

Target audience: Software developers and testers as well as their technical managers who want to understand and use Github Copilot and spec-kit in agentic coding and spec-driven software development.

Prerequisites: Experience with VSCode or similar IDEs, experience in using ChatGPT or similar chatbots, basic knowledge of the programming language used during lab exercises.

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

- AI-101: Intro to GenAI with Large Language Model (LLMs) and LLM-based apps.
- AI-141: Using GitHub Kiro as coding assistant
- AI-161: Using Amazon Q as coding assistant
- AI-242: Using Github Copilot and spec-kit for agentic coding and spec-driven development (SDD)
- AI-262: Using Amazon Kiro for agentic coding and spec-driven development (SDD)
- AI-434: GenAI Application Development with LLMs
- AI-452: Agentic AI Application Development with LLMs