Responsible AI in agentic software development (RAIASD)

ID RAIASD **Price** on request **Duration** 1 day

Who should attend

All people involved in using agentic AI tools in software development

Prerequisites

General development

Course Objectives

- Understand various aspects of responsible AI
- How to use generative AI responsibly in software development
- Prompt engineering for optimal outcomes
- How to apply generative AI throughout the SDLC
- The challenges in using agentic GenaI

Course Content

A brief history of Artificial Intelligence

- The origins of AI
- Neural networks and “probability engines”
- Early ML coding tools
- The AI coding revolution of the 2020s

Responsible AI

- What is responsible AI?
- Accountability and transparency
- Mitigation of harmful bias
- Validity and reliability
- Demonstration – Experimenting with validity and reliability in Copilot
- Explainability and interpretability
- Safety, security, privacy and resilience
- Security and responsible AI in software development

Using GenAI responsibly in software development

- LLM code generation basics
- Basic building blocks and concepts
- Prompt templating
- System prompts in AI-driven coding
- Can AI… boost your productivity?
- Can AI… take care of the ‘boring parts’?
- Can AI… be more thorough?
- Reviewing generated code – the black box blues
- The danger of hallucinations
- The effect of GenAI on programming skills
- Some further long-term effects of using GenAI
- Where AI code generation doesn’t do well
- Prompt engineering
    - Why is a good prompt so important?
    - Establishing the context for generative AI
    - Zero-shot, one-shot, and few-shot prompting
    - Reasoning-based prompt engineering, chain-of-thought
    - Demonstration – Experimenting with prompts in Copilot
    - Enforcing and following token limits
    - Prompt patterns
        - Prompt patterns and prompt priming
        - The 6 categories of prompt patterns
    - Some further prompting approaches
        - Least-to-Most and Self-Planning: decomposition of complex tasks
        - Demonstration – Task decomposition with Copilot
        - Unit tests, TDD and GenAI
        - Demonstration – Test-based code generation with Copilot
- Integrating generative AI into the SDLC
    - Using GenAI beyond code generation
    - Using AI during requirements specification
    - Prompt patterns for requirements capturing
    - Prompt patterns for software design
    - Demonstration – Requirements capturing and API design with Copilot
    - Using AI during implementation
    - Prompt patterns for implementation
    - Demonstration – Finding hidden assumptions with Copilot
    - Using AI during testing and QA
- Agentic software development
    - Intelligent agents and GenAI
        - How is agentic coding different?
        - The Model Context Protocol (MCP)
        - Capabilities of MCP agents

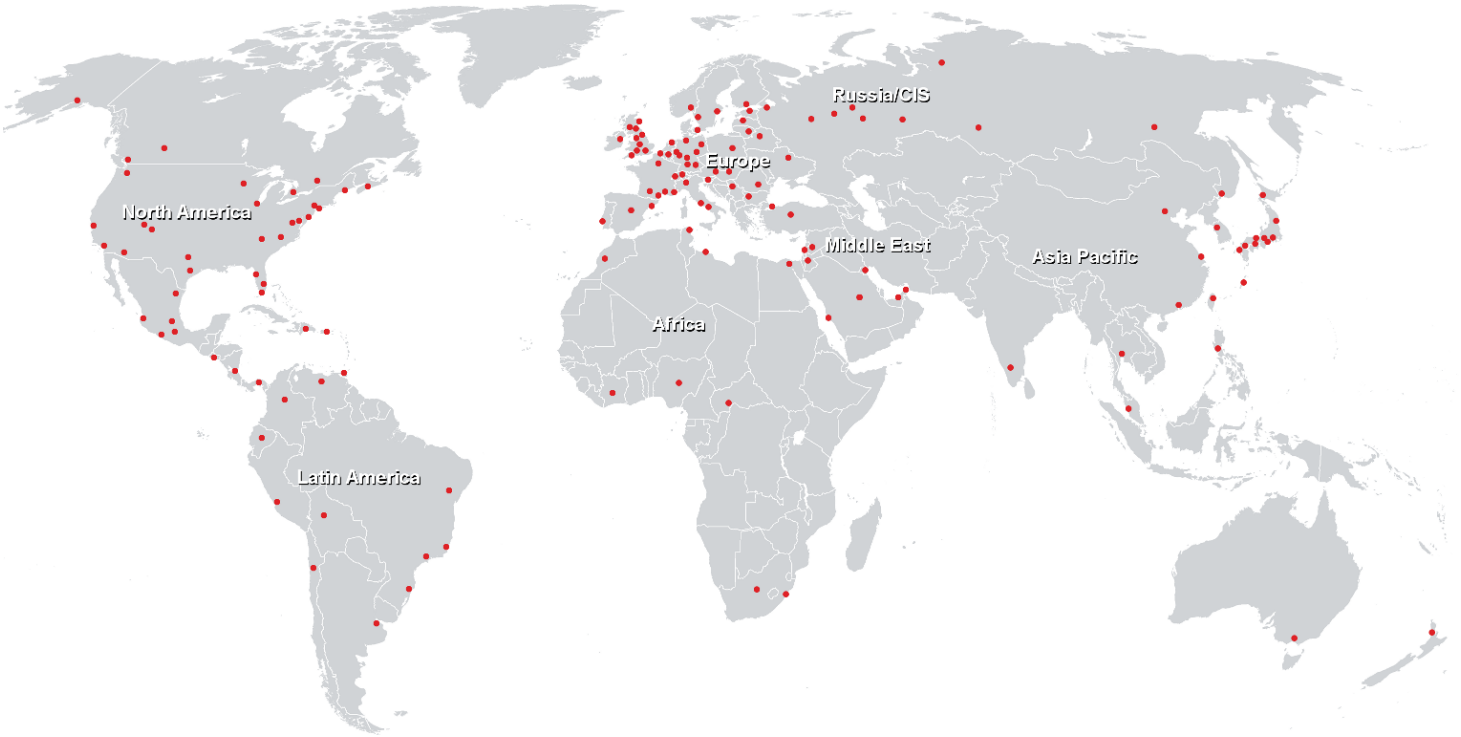
        - Agentic integration in IDEs
    - Agentic development workflow
        - Code-to-spec and spec-to-code with GenAI
        - Automated scaffolding
        - Demonstration – Agentic scaffolding with Copilot
        - Setting up the runtime environment
        - Demonstration – Environment setup with Copilot
        - Incremental development
        - Demonstration – Incremental development with Copilot
        - The role of MCP in Dev(Sec)Ops
        - Demonstration – Using MCP in DevOps with Copilot
    - Pitfalls and best practices
        - “Vibe coding” and its implications
        - Engineering concerns with MCP
        - Security concerns of agentic development
        - MCP’s effect on the attack surface
        - MCP-specific attack vectors
        - Demonstration – Attacking agentic Copilot
        - Case study – Database leakage via Supabase MCP
        - Hallucinations and ‘agentic death spirals’
        - Token limits and context
        - Context degradation with very large token counts
        - Prompt engineering vs context engineering
        - Context engineering from a developer’s perspective
        - Context document examples

Summary and takeaways

- Responsible AI principles in software development
- Generative AI – Resources and additional guidance

Responsible AI in agentic software development (RAIASD)

Training Centres worldwide



Fast Lane Institute for Knowledge Transfer (Switzerland) AG

Husacherstrasse 3
CH-8304 Wallisellen
Tel. +41 44 832 50 80

info@flane.ch, https://www.flane.ch