

Vibe Coding Training

Duration: 2 Days

Description

The purpose of this training is to enable participants to effectively leverage generative AI tools within modern software development processes while ensuring efficiency, governance, and alignment with enterprise standards.

The training provides a practical framework for integrating AI into the software development lifecycle and helps teams adopt modern AI-assisted development practices in a structured and responsible way.

By the end of the training, participants will be able to manage AI-supported development workflows end-to-end, utilize modern AI coding tools, and establish a sustainable AI usage model within their organizations.

Key Learning Outcomes

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

- Design and implement an enterprise-ready AI-assisted software development approach
- Develop a standardized organizational guideline for AI usage in development
- Build and manage an effective prompt library for development teams
- Integrate AI into testing, debugging, and refactoring workflows
- Understand token management and AI cost optimization strategies

Outline

Day 1 - Fundamentals and the Modern AI Coding Ecosystem

AI and LLM Fundamentals

- Generative AI and Large Language Model (LLM) concepts
- How LLMs work (high-level architecture)
- Token, context window, and embedding concepts
- Hallucination issues and reliability limitations

- Differences between RAG and fine-tuning approaches
- Setting the right expectations for AI-assisted software development

Token Usage and Context Management - Best Practices

- Incremental development approach (micro-tasking)
- Avoiding unnecessarily verbose prompts
- Using summarization to maintain clean context
- Creating prompt templates and a prompt library
- Defining metrics for token usage and cost monitoring

Modern AI Coding Tools

- Cursor
 - Agent Mode
 - Project-level context management
 - Refactoring and diff analysis
- Claude Code
 - Terminal-based AI coding
 - Advantages of extended context capabilities
- Antigravity
 - Repository-level analysis
 - Architecture exploration
- VS Code + LLM Integrations
 - GitHub Copilot
 - Agent-based development workflows

Day 2 - Advanced Usage and Enterprise Adoption

AI for Testing, Refactoring, and Debugging

- Generating unit tests and edge cases with AI
- Refactoring and clean code practices

- Performance optimization scenarios
- AI-assisted error analysis and root cause identification

Security and Enterprise Compliance

- Verifying AI-generated code
- Identifying potential security vulnerabilities
- PII and data security considerations
- On-premise vs. cloud model usage strategies
- Logging and audit approaches

Team-Based AI Usage Model

- Establishing an AI Coding Policy
- Creating a Prompt Usage Guide
- Integrating AI into code review processes
- AI-assisted sprint planning
- Implementing an AI-driven development squad model