

Updated on 06/01/2026

Sign up

Agent-Based Engineering Training: From Code to Orchestration

3 days (21 hours)

Overview

Most teams use AI as glorified autocomplete—without architecture, without governance, and without a method to validate its output. The result: technical debt piling up at an industrial pace.

This training course does not focus on quick fixes. It trains conductors: designing multi-agent systems, optimizing inference costs, managing context as a critical resource, and validating what the agent produces rather than coding line by line.

The common thread: reduce your build time by a factor of three to focus 100% on product value. The concepts are framework-agnostic—applicable regardless of the stack.

Objectives

- Adopt this new level of abstraction
- Delegate work while maintaining control: validate the relevance of generated code, anticipate side effects, and correct hallucinations in real time
- Secure an agent-based workflow in the enterprise: data confidentiality, masking, local models vs. remote APIs

Target Audience

- Intermediate and senior developers looking to transition from writing code to orchestrating AI systems, regardless of their primary language
- Tech leads and staff engineers responsible for architecture and quality on projects with agent-based components
- Engineers who have already integrated an LLM API into production and are facing issues with cost, reliability, or maintainability

- Teams looking to introduce AI into an enterprise environment with real security and compliance constraints

Prerequisites

- Operational proficiency in at least one programming language: API handling, code structuring
- Experience with version control (Git or equivalent): branches, history, collaboration
- Experience using at least one LLM API (OpenAI, Claude, or equivalent) in a personal or professional project
- Minimum experience with one production project involving a backend component

Technical prerequisites

- Functional local development environment, regardless of the language used
- Account on a version control platform (GitHub, GitLab, or equivalent) with configured access
- Active API key from an LLM provider (Claude, OpenAI, or equivalent)

Agent Engineering Training Program: From Code to Orchestration

[Day 1 - Morning]

Developing with AI: Philosophy Before Tools

- Paradigm Shift: Debunking Myths, Laying the Groundwork — Mindset & Method
- From Code to Orchestration: The Engineer's New Role — Approach & Method
- Multi-agent architecture: routing, memory, feedback loops — Architecture

[Day 1 - Afternoon]

Developing with AI: Philosophy Before Tools (continued)

- The agent-based glossary: vocabulary as a tool for precision — Vocabulary
- Under the hood of LLM: understanding to better manage — Architecture
- Tools and MCP: selection of useful skills in an enterprise context — Tools
- Hands-on workshop

[Day 2 - Morning]

Architecture & Quality: Framing the agent to produce maintainable code

- The reviewer's approach: validate, anticipate side effects, correct — Approach & Method
- Context management: conserving tokens, cost as a design constraint — Architecture
- Clean architecture & the power of concepts: Intent? Orchestration? Execution? Data — Architecture

[Day 2 - Afternoon]

Architecture & Quality: Framing the Agent to Achieve Maintainable Code (continued)

- Design patterns as a specification language — Facade, Strategy,
- Observer, Factory, Command — Architecture
- Agent-based TDD: writing the contract before letting the agent implement — Method
- The 6 golden rules: precisely framing the agent — Approach & method
- RAG & vector databases: anchoring agents in domain knowledge — Architecture
- Practical workshop

[Day 3 - Morning]

Security, teamwork & deployment

- Security & compliance: local models vs. APIs, data masking, AI reverse proxy — Security
- Team governance: adoption policy, transparent boundaries, non-determinism — PM & process
- Lab: Launching the team exercise — briefing, forming groups, defining the scope of the case

[Day 3 - Afternoon]

Security, teamwork & deployment (continued)

- Lab: Team sprint — design, code, and secure a complete agent-based system
- Lab: Presentation & peer review — each team defends its architectural and security choices

Target Audience

This training is intended for both individuals and companies, large or small, wishing to train their teams in a new advanced IT technology or to acquire specific business knowledge or modern methods.

Entry-Level Assessment

The pre-training assessment complies with Qualiopi quality standards. Upon final registration, the learner receives a self-assessment questionnaire that allows us to evaluate their estimated proficiency in various types of technologies, as well as their expectations and personal goals for the upcoming training, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could pose challenges for monitoring and ensuring the smooth running of the training session.

Teaching Methods

Practical Training: 60% Practical, 40% Theoretical. Training materials will be distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the instructor, supported by examples and reflection sessions, and group work.

Assessment

At the end of the session, a multiple-choice questionnaire is used to verify that the skills have been properly acquired.

Certification

A certificate will be issued to each trainee who has completed the entire training program.