

AI Agent Architecture and Development Training Course

4 – 8 January 2027
London

AGILE LEADERS
Training Center

AGILE LEADERS
Training Center

AI Agent Architecture and Development Training Course

Ref: 198_193295 **Date:** 4 – 8 January 2027 **Location:** London **Fees:** 5700 Euro

Overview

AI Agent Architecture and Development Training Course is a one-week intermediate course for developers, solution architects, automation engineers, AI product specialists, and technical leads, who leave with a Tested AI Agent Prototype. Participants design agent architecture, goal and task decomposition, tool contracts, retrieval and memory boundaries, orchestration patterns, evaluation, guardrails, security, observability, and deployment readiness. Agile Leaders Training Center delivers training in practical AI agent engineering.

Who Should Attend

- Software development personnel responsible for intelligent applications and service integration
- Solution architecture personnel responsible for system boundaries, interfaces, and deployment choices
- Automation personnel responsible for multi-step workflows, tools, and exception handling
- AI product personnel responsible for use cases, acceptance criteria, and release decisions
- Technical leadership personnel responsible for engineering quality, security, and delivery readiness

The course assumes participants can read or write application logic and work with service interfaces, and it leaves out introductory AI literacy, model training, and purely executive governance.

Departments and Industries

The course supports technical delivery across software, financial services, health care, retail, logistics, public services, and professional services.

- Software engineering and solution architecture
- Automation and platform engineering
- Data, AI, and product development
- Cybersecurity and technology risk
- Digital services across finance, health care, retail, logistics, and government

Learning Objectives

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

- Analyze agent use cases and architectural boundaries
- Build goal, task, tool, retrieval, and memory components
- Apply orchestration and multi-agent interaction patterns
- Evaluate trajectories, outputs, failures, and recovery behavior
- Use guardrails, access controls, traces, and human review
- Build and test a deployment-ready agent prototype

Course Agenda

Day 1: Agent Architecture and Task Design

- Agent Use-Case Suitability Matrix
- Goal and Success-Criteria Canvas
- Agent Loop Architecture Diagram
- Task Decomposition and State Model
- Human Authority and Escalation Map

Day 2: Tools, Retrieval, and Memory

- Tool Contract and Permission Schema
- Tool-Selection and Error-Handling Flow
- Retrieval Source and Evidence Boundary
- Session and Long-Term Memory Design
- Context Budget and State Checklist

Day 3: Orchestration and Multi-Agent Patterns

- Sequential Orchestration Pattern
- Handoff and Routing Decision Table
- Coordinator-and-Specialist Agent Topology
- Parallel Task and Result Synthesis Flow
- Agent Communication Trust Boundary

Day 4: Evaluation, Security, and Operations

- Agent Trajectory Evaluation Rubric
- Tool-Use Test Case Library
- Guardrail and Human-Review Control Map
- Trace-Based Observability Dashboard
- Deployment Readiness and Rollback Gate

Day 5: Agent Build Practice and Capstone

- Suggested Exercise: Model an Agent Task and State Loop
- Suggested Exercise: Implement and Test a Tool Contract
- Suggested Exercise: Diagnose a Trace and Recovery Path
- Suggested Exercise: Evaluate Security and Deployment Gates
- Capstone Exercise: Tested AI Agent Prototype

Practical Exercises

The course uses suggested activities that convert an agent use case into a testable technical design and prototype.

- Suggested activity: define goals, success criteria, task states, authority limits, and escalation for a service or operations use case
- Suggested activity: specify tool inputs, outputs, permissions, errors, retrieval evidence, and memory retention boundaries
- Suggested activity: compare sequential, routed, parallel, and coordinator patterns for a multi-step workflow
- Suggested activity: run trajectory tests, inspect traces, challenge guardrails, and make a release or rollback decision

FAQs

Who suits the AI Agent Architecture and Development Training Course, and what does it assume?

The course suits developers, architects, automation engineers, AI product specialists, and technical leads. It assumes familiarity with application logic and service interfaces rather than prior expertise in a specific agent framework.

How does AI agent development training differ from introductory generative AI training?

AI agent development training focuses on architecture, tools, state, memory, orchestration, evaluation, security, and deployment, while introductory training focuses on concepts and direct model interaction.

What components belong in an AI agent architecture?

An AI agent architecture typically defines goals, instructions, task state, model interaction, tool interfaces, retrieval, memory, orchestration, guardrails, human review, traces, evaluation, and operational controls.

How should technical teams evaluate AI agents?

Technical teams should test task success, execution trajectories, tool choices, evidence use, failure recovery, security boundaries, latency, cost behavior, human escalation, and consistency across representative cases.

What makes an AI agent ready for deployment?

Deployment readiness requires defined ownership, tested tools and permissions, bounded memory, evaluation evidence, observable traces, guardrails, human escalation, failure recovery, release criteria, monitoring, and rollback procedures.

Conclusion

Participants take back a Tested AI Agent Prototype containing an architecture, tool contracts, memory boundaries, orchestration logic, evaluation cases, guardrails, traces, and deployment gates. It changes how technical teams move from an agent idea to controlled implementation. The prototype supports repeatable testing, security review, operational diagnosis, and evidence-based release decisions.