

Updated on 07/22/2026

Sign Up

Expert Agent Development Training with CrewAi and Flowise

4 days (28 hours)

Overview

Agent-based development and generative AI are redefining modern software architecture. This training course takes a Hybrid Anchored approach, combining programmatic power (Python and CrewAI) with visual acceleration (Flowise's no-code platform) to design intelligent and autonomous systems.

Our Generative AI and Agent-Based Development training will enable you to master advanced prompt engineering, build complex data pipelines for RAG, and model autonomous behaviors through the orchestration of multi-agent systems.

You'll also learn to design hybrid architectures, secure your AI applications (Guardrails), optimize inference costs (FinOps), and deploy these systems into production at scale.

Upon completion of the course, you will be able to independently design, deploy, and maintain complex multi-agent systems and RAG pipelines tailored to enterprise security and performance requirements.

Like all our training courses, this one will introduce you to **the latest stable version** of the technology and its new features.

Objectives

- Master advanced techniques in prompt engineering and output structuring (JSON/Pydantic).
- Design and deploy a complete and optimized RAG (Retrieval-Augmented Generation) pipeline.
- Develop autonomous agents and orchestrate multi-agent systems with CrewAI.

- Orchestrate generative AI workflows using the Flowise visual and node-based interface.
- Secure, monitor (observability), and scale an AI application in production.

Target Audience

- Senior developers
- Tech Leads
- Software architects

Prerequisites

- Advanced proficiency in Python programming
- Solid knowledge of software architecture and REST API calls
- Familiarity with Git and containerized architectures (Docker)

Technical Requirements

- At least 16 GB of RAM (32 GB recommended for smooth execution of local containers)
- Linux, macOS, or Windows (WSL2 must be configured)
- Python 3.10+ and Node.js 18+ installed on the machine
- Docker and Docker Compose installed with administrator privileges
- An advanced code editor (VS Code, PyCharm)

Curriculum for our Expert Agent Development Training with CrewAi and Flowise

[Day 1 - Morning]

Fundamentals of Generative AI and Advanced Prompt Engineering

- The No-Code Paradigm for Experts: Prototyping Accelerator, Productivity Gains, and Integration Patterns
- Advanced Prompt Engineering: Few-Shot Techniques, Chain-of-Thought (CoT), ReAct (Reasoning and Acting), and Directional Stimulus
- Determinism and Output Structuring: Mastering JSON Mode, Structured Outputs, and Parsing via Pydantic Data Schemas
- Industrial-Grade Chain Architectures and Prompting Frameworks
- Ecosystem Management: LLMs, Context Windows, and Tokenization
- Hands-On Workshop: Designing complex programmatic prompts with strict validation of output data structures.

[Day 1 - Afternoon]

AI Product Architecture & Visual Orchestration with Flowise

- Anatomy of a Modern AI-Powered Application: From the User Interface to LLM Calls
- Introduction to Flowise: Node-based interface for developers, concepts of components, chains, and graphs
- Creating simple component chains and advanced conversational agents (chatbots, voicebots)
- State and Persistence Management: Advanced Memory Strategies (Buffer Window, Redis Memory) and User Session Persistence
- Scaling: Exposing Flowise flows as secure REST APIs
- Hands-on Workshop: Deploying a visual conversational agent (advanced chatbot) connected to a third-party webhook.

[Day 2 - Morning]

Data Pipelines and Vectorization for RAG

- RAG (Retrieval-Augmented Generation): Principles, use cases vs. fine-tuning, and reference architecture
- Data Pipelines (ETL for LLM): Collection, cleaning of unstructured data, and semantic enrichment
- Chunking Strategies: Recursive, semantic, and document-structure-based approaches (Markdown, Code)
- Embedding Models and Indexing: Model Selection, Dimensions, and (Cosine, L2, Dot Product)
- Overview of vector databases: Technical comparison (Pinecone, Qdrant, Chroma, PGVector)
- Hands-on Workshop: Writing a script for automated ingestion and indexing of technical documentation.

[Day 2 - Afternoon]

Implementation and Optimization of RAG

- Implementing a Complete End-to-End RAG Pipeline in Flowise
- Advanced search techniques: Hybrid Search (Dense + Sparse / BM25) and tuning relevance scores
- Optimizing Results: Integrating re-ranking models (Cohere, FlashRank) to maximize useful context
- Evaluation and Maintenance: Measuring RAG accuracy, reducing hallucinations, and managing incremental updates to the vector database
- Hands-on workshop: Building a secure enterprise RAG using the learners' own data learners.

[Day 3 - Morning]

Agent Development: Architecture and Foundations of CrewAI

- Anatomy of an Autonomous Agent: The Pillars of the Agent (Perception, Cognitive Reasoning Cycles, Action via Tools)
- Introduction to CrewAI: A Role-Based Framework. Configuring Roles, Goals (Goals) and Backstories
- Defining Tasks: Breaking Down a Business Requirement into Agent Deliverables
- Creating Custom Tools: Python Decorators, Encapsulation of Existing APIs, and Agent Exception Handling
- Context Management and History Passing Between Tasks
- Hands-on Workshop: Complete Python implementation of an autonomous search agent equipped with custom scraping and analysis tools.

[Day 3 - Afternoon]

Multi-Agent Systems: Orchestration and Collaboration

- Multi-Agent Orchestration Patterns: Sequential, Hierarchical, and Iterative Processes
- Delegation and Supervision: Managing Virtual Managers and Inter-Agent Communication in CrewAI
- Agent Memory Mechanisms: Management of short-term, long-term, and shared memory among members of the same "Crew"
- Architectural Comparison: Advantages and Limitations of CrewAI Compared to State Graph Architectures (LangGraph)
- Hands-on Workshop: Creating a "Crew" of virtual engineers (Product Owner, Developer, QA) who collaborate to generate, test, and validate source code.

[Day 4 - Morning]

Security, Guardrails, and Industrialization

- Vulnerabilities of Autonomous Agents: Prompt Injections (Direct/Indirect), Data Leaks, and Infinite Execution Loops
- Implementing security barriers: Using guardrail frameworks (Guardrails, Llama Guard)
- FinOps and Robustness: Monitoring token costs, implementing rate limiting, and managing API timeouts
- Agent Observability: Comprehensive tracing of decision trees with tools such as LangSmith or Phoenix
- Hands-on workshop: Hardening agent security and setting up a dashboard for tracing LLM calls.

[Day 4 - Afternoon]

Final Case Study: Deploying an Autonomous Multi-Agent System

- Final Project Brief: Design a complex system combining an RAG pipeline (Flowise) and multi-agent business logic (CrewAI)
- Solution Architecture: Hybrid modeling (Flowise for the interface/RAG, CrewAI for business logic and orchestration)
- Development under supervised guidance, error handling (fallbacks), and system resilience
- Technical Acceptance Testing and Deployment: Dockerization of the solution and preparation for production
- Hands-on Workshop / Defense: Finalization, deployment, demonstration of the designed systems, and peer code reviews.

[Technical Details & Deliverables]

- System Requirements (14 days prior): The software and hardware configuration must be finalized at least 2 weeks before the start of the project.
- End of Service: Provision of comprehensive documentation (sample architectures, workshop and project code).

Target Companies

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

Assessment upon enrollment

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

Hands-On Training: 60% Practical, 40% Theoretical. Training materials 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 quiz is administered to verify that participants have successfully acquired the skills.

Certification

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