www.logicalimagination.com
800.657.1494

AI Architecture & Agentic Systems

Course #: AI-202 **Duration:** 2 days

Prerequisites

Completion of AI Foundations & Risk for IT Professionals and Designing Reliable AI Workflows & Interactions, or equivalent experience designing AI-enabled workflows with an understanding of risk and governance.

Details

This course focuses on the architecture of agentic AI systems in which AI models act as autonomous or semi-autonomous actors that plan, reason, and interact with tools, data, and workflows.

Participants learn how agentic systems differ fundamentally from chat-based AI, why naïve agent designs fail, and how architectural discipline enables safety, reliability, and control. Emphasis is placed on **boundaries, authority, memory, orchestration, and failure containment**, preparing participants for hands-on implementation using structured protocols and tooling.

After attending this course, students should be able to:

Explain the defining characteristics of agentic AI systems
Distinguish between chat-based AI, workflow-based AI, and agentic architectures
Identify core architectural components of agentic systems
Recognize common agent failure modes and design flaws
Apply architectural principles that support safe, bounded AI autonomy
Evaluate agentic system designs for reliability, risk, and maintainability

This course is designed for IT professionals responsible for designing, evaluating, or overseeing advanced AI system architectures, including architects, senior engineers, and technical leaders. This course is technical and architectural in nature but does not require prior agent or MCP implementation experience.

Software Needed

A laptop or desktop computer with a modern web browser and reliable internet access is recommended for accessing course materials and participating in discussions. No programming tools or AI software access is required.

Outline

AI Architecture & Agentic Systems

- **From AI Tools to AI Actors**
  - Limitations of chat-based AI systems
  - Why autonomy changes system risk
  - The shift from “response generation” to “action execution”
  - Real-world examples of agentic behavior

- **What Makes a System Agentic**
  - Defining autonomy, goals, and decision loops
  - Planning vs execution
  - Internal vs external control
  - Degrees of autonomy and supervision
- **Core Components of Agentic Architectures**
  - The agent core (reasoning and planning)
  - Tools, actions, and external systems
  - Memory and state management
  - Orchestration and coordination
- **Authority, Boundaries, and Scope**
  - Defining what an agent is allowed to do
  - Preventing overreach and unintended actions
  - Scoping authority to roles and tasks
  - Designing safe defaults
- **Memory, Context, and State**
  - Short-term vs long-term memory
  - State persistence and recovery
  - Context drift and contamination
  - Strategies for memory isolation
- **Control Loops and Supervision**
  - Planning, acting, observing, and adjusting
  - Human-in-the-loop vs human-on-the-loop
  - Monitoring agent behavior
  - Interrupting and stopping agents safely
- **Common Failure Modes in Agentic Systems**
  - Runaway behavior and infinite loops
  - Hallucinated plans and actions
  - Tool misuse and incorrect assumptions
  - Compounding errors across steps
- **Designing for Safety and Reliability**
  - Containment and blast-radius reduction
  - Idempotency and repeatability
  - Auditability and traceability
  - Aligning agent behavior with governance
- **Multi-Agent and Orchestrated Systems**
  - When multiple agents are appropriate
  - Coordination and communication patterns
  - Shared vs isolated state
  - Failure propagation across agents
- **Evaluating Agentic System Designs**
  - Architectural review criteria
  - Risk assessment for autonomy
  - Readiness for implementation
  - Red flags and anti-patterns
- **Preparing for Structured Implementation**
  - Why ad-hoc agents fail in production
  - The role of protocols and schemas
  - Transitioning to MCP-based systems
  - Preparing for hands-on agent implementation
- **Summary and Next Steps**
  - Key architectural principles
  - Applying agentic thinking responsibly
  - Aligning architecture with organizational maturity
  - Moving into MCP-based system design and development