**AI Agent:** follow workflow that combines perception, reasoning, and action.

- Autonomous s/w system that

- perceives its environments

- process information,

- make decisions

- take actions to achieve specific goals,

- improving its performance through learning or adaptation

**Goal-Based Agents**: Plan actions to achieve specific objectives. Example: A travel agent booking optimal itineraries.

**Decide**:

* Will your agents be **assistive**, **autonomous**, or **collaborative**?
* Will they work with humans (co-pilots) or operate independently (auto-pilots)?
* What levels of decision-making authority will they have?

📌 **Outcome**: **A taxonomy of agentic roles and interaction** modes.

**What do they have?**

* Memory
* Tools/skill
* Router

| **Component** | **Description** | **Example** |
| --- | --- | --- |
| **Perception** | Gathers data from the environment via sensors, APIs, or user inputs to understand context. | A customer service agent analyzing query history and real-time data. |
| **Reasoning/Planning** | Uses AI models (e.g., LLMs) to analyze information, decompose tasks, and decide on actions. | Breaking a travel planning goal into subtasks like checking weather and booking flights. |
| **Memory** | Stores short-term (e.g., current session) and long-term data for learning and adaptation. | Recalling past interactions to personalize responses. |
| **Tools/Actions** | Integrates external resources like web searches, databases, or APIs to execute tasks. | Calling an API to process a payment or search for information. |
| **Learning/Feedback** | Improves via reinforcement learning, user feedback, or self-reflection to refine future performance. | Iteratively adjusting based on successes or failures. |

Agentic systems, often referring to AI agents capable of autonomous and goal-directed behavior, rely on a set of foundational capabilities that enable them to operate effectively in dynamic environments. Based on analyses from various industry and expert sources, these core capabilities include:

- \*\***Autonomy**\*\*: The ability to operate independently, initiating and completing tasks with minimal or no human intervention.

- \*\***Goal-Orientation**\*\*: Pursuing predefined or evolving objectives, including setting and prioritizing goals to achieve desired outcomes.

- \*\***Reasoning and Decision-Making\*\*:** Evaluating contexts, trade-offs, and options to make informed decisions, often leveraging sophisticated logic or models.

- **\*\*Planning\*\*:** Breaking down high-level goals into subtasks, sequencing actions, and dynamically adjusting plans based on new information.

- **\*\*Memory and Contextual Awareness**\*\*: Retaining information from past interactions, maintaining context over time, and incorporating it into ongoing processes.

- \*\***Adaptability and Self-Learning\*\*:** Adjusting to changes, learning from feedback or experiences, and improving performance through reinforcement or iterative processes.

- **\*\*Perception and Environment Awareness\*\*:** Sensing and interpreting the surrounding environment, including real-time data ingestion and analysis.

- **\*\*Action-Taking and Execution**\*\*: Proactively performing tasks, optimizing workflows, and interacting with external systems or tools to achieve results.

- **\*\*Interactivity and Collaboration**\*\*: Communicating with humans, other agents, or systems, including language understanding and multi-agent coordination.

| **Category** | **Example Capabilities** | **Why Foundational for Wells Fargo?** |
| --- | --- | --- |
| **Autonomy & Action** | - Independent transaction processing and fraud alerts - Goal-oriented workflows for loan approvals - Adaptive responses to market fluctuations | Enables proactive banking operations, such as real-time fraud intervention, aligning with Wells Fargo's focus on secure, efficient services. Integrates with Google Agentspace for seamless execution. |
| **Reasoning & Learning** | - Contextual risk analysis using historical data - Reinforced learning for personalized financial advice - Memory for customer interaction history and reflection on outcomes | Supports intelligent decision-making in volatile markets, improving wealth management and compliance. Builds on NotebookLM for document-based reasoning. |
| **Interaction & Integration** | - API/tool use for core systems (e.g., integrating with payment gateways or CRM) - Natural language processing for customer queries - Multi-agent collaboration (e.g., fraud agent + compliance agent) | Facilitates ecosystem-wide deployment, as in Wells Fargo's rollout of AI agents for customer service and market insights. Uses Google Cloud's A2A protocol for agent communication. |
| **Governance & Security** | - Regulatory compliance checks (e.g., AML/KYC validation) - Bias mitigation in lending decisions - Auditing and logging for all actions, with encryption for sensitive data | Critical for financial trust; prevents risks like data breaches or biased outcomes. Aligns with Wells Fargo's foundational platforms for safe AI, including ethical constraints. |

**5. Validate, Iterate, and Scale**

* Validate: Measure KPIs like response time in customer interactions, accuracy in fraud detection, or ROI from automation.
* Continuous learning: Update based on feedback and AI advancements, ensuring agents adapt to new regulations (e.g., post-2025 CFPB rules).
* Scale: Document in a Wells Fargo AI playbook, train employees via internal platforms, and expand business-wide as per the Google collaboration.
* Action: Quarterly reviews, integrating with enterprise governance to address challenges like legacy system integration.

**Reusable Patterns for Agentic Systems**

| **Pattern Name** | **Description** | **Use Cases** | **Benefits** |
| --- | --- | --- | --- |
| **Reflection** | The agent evaluates its own outputs iteratively, using feedback (e.g., critiques from another LLM) to refine responses until criteria are met. | Code generation with bug fixing; content refinement for quality. | Improves accuracy and reliability through self-correction. |
| **Tool Use** | Agents invoke external tools, APIs, or functions to gather data or perform actions, with the LLM deciding when and how to use them. | Real-time data retrieval (e.g., stock prices via API); smart home control. | Extends agent capabilities beyond internal knowledge. |
| **Planning** | A central agent decomposes complex goals into subtasks, creates dynamic plans, and iterates based on outcomes. | Trip planning (e.g., budgeting, booking); software development workflows. | Handles uncertainty and long-horizon tasks effectively. |
| **Control Plane as Tool** | Exposes a single tool interface to the agent while hiding modular routing logic, enabling scalable tool orchestration. | Enterprise integrations where agents access multiple backend services via one endpoint. | Simplifies agent design; enhances modularity and security. |
| **Pattern Name** | Description | Use Cases | Benefits |
| **Multi-Agent Collaboration** | Multiple agents with specialized roles interact to solve tasks, often coordinated by a manager agent. | Debates for decision-making; software creation (planning, coding, testing agents). | Leverages diverse expertise; scales for complex problems. |
| **Orchestrator-Worker** | A central orchestrator delegates subtasks to worker agents, collects results, and synthesizes outputs. | Research reports (e.g., data gathering workers feeding an analyzer). | Centralizes control; efficient for hierarchical tasks. |
| **Hierarchical** | Agents organized in layers, with higher-level agents supervising lower ones for decision-making and delegation. | Enterprise management (e.g., executive agent overseeing department agents). | Supports scalability in large organizations. |
| **Blackboard** | Agents contribute to a shared "blackboard" space for knowledge exchange, without direct communication. | Collaborative problem-solving like brainstorming sessions. | Decentralized; reduces communication overhead. |
| **Market-Based** | Agents "bid" or negotiate resources/tasks based on incentives, mimicking economic models. | Resource allocation in dynamic environments (e.g., cloud computing bids). | Optimizes efficiency through competition. |
| **Group Chat** | Agents engage in conversational exchanges, voting or debating to reach consensus. | Multi-perspective analysis (e.g., risk assessment teams). | Fosters emergent intelligence from interactions. |
| **Handoff** | One agent passes control to another based on expertise or task phase. | Customer service escalation from bot to specialist. | Ensures smooth transitions; maintains continuity. |

**Standardize Each Pattern:**

Create a reusable template:

* **Pattern Name**
* **Problem it Solves**
* **Context of Use**
* **Structure / Flow Diagram**
* **Example Code / Prompt**
* **Failure Modes / Caveats**

**🧰 Tools to Use:**

* **Prompt engineering repositories**
* **LangChain / Semantic Kernel templates**
* **OpenAI function / tool call registries**
* **Architecture diagrams (e.g., Mermaid, draw.io)**

**📦 Store in a Shared System:**

Put these in:

* A Notion or Confluence wiki
* GitHub repo with templates
* Internal design system for agents

1. Inspection agent - > all use cases from Jithin as each tool