AI Masterclass

Technical Generative AI Concepts Explained Simply

github.com/Esteban-D-Lopez/TheAIWalkthrough

Learning Journey Roadmap

01 Technical Generative AI Foundations

Introduce foundational technical knowledge about AI and large language models (LLMs), laying the groundwork for understanding Generative AI.

02 GenAI Optimization Techniques 1

An overview of key LLM optimization techniques, with a deep dive into Fine-Tuning and Prompt Engineering.

03 GenAI Optimization Techniques 2

An overview of key LLM optimization techniques, with a deep dive into Retrieval Augmented Generation (RAG) and Agentic AI.

04 Generative AI Monitoring and Evaluation

An overview of key implications and practical considerations of bringing Generative AI products to life safely and efficiently.

Goals

✓ Understand how GenAI technology works

✓ Feel comfortable exploring with GenAI tools

✓ Start applying GenAI technology safely and responsibly

AI Masterclass

Presented by

Esteban D. Lopez

Location	New York, NY
Hometown	Quito, Ecuador
Education	University of New Orleans – BS. Accounting Louisiana State University – MS. Accounting Columbia University – FinTech Bootcamp New York University – MS. AI and Machine Learning (Current)
Professional Career	Hedge Funds Assurance Transformation & Economic Consulting AI & Data Technology Consulting AI Product Management

Learning Journey Roadmap

● 01 **Technical Generative AI Foundations**

Introduce foundational technical knowledge about AI and large language models (LLMs), laying the groundwork for understanding Generative AI.

● 02 **GenAI Optimization Techniques 1**

An overview of key LLM optimization techniques, with a deep dive into Fine-Tuning and Prompt Engineering.

● 03 **GenAI Optimization Techniques 2**

An overview of key LLM optimization techniques, with a deep dive into Retrieval Augmented Generation (RAG) and Agentic AI.

● 04 **Generative AI Monitoring and Evaluation**

An overview of key implications and practical considerations of bringing Generative AI products to life safely and efficiently.

Goals

- ✓ Understand how GenAI technology works
- ✓ Feel comfortable exploring with GenAI tools
- ✓ Start applying GenAI technology safely and responsibly

Chapter 3

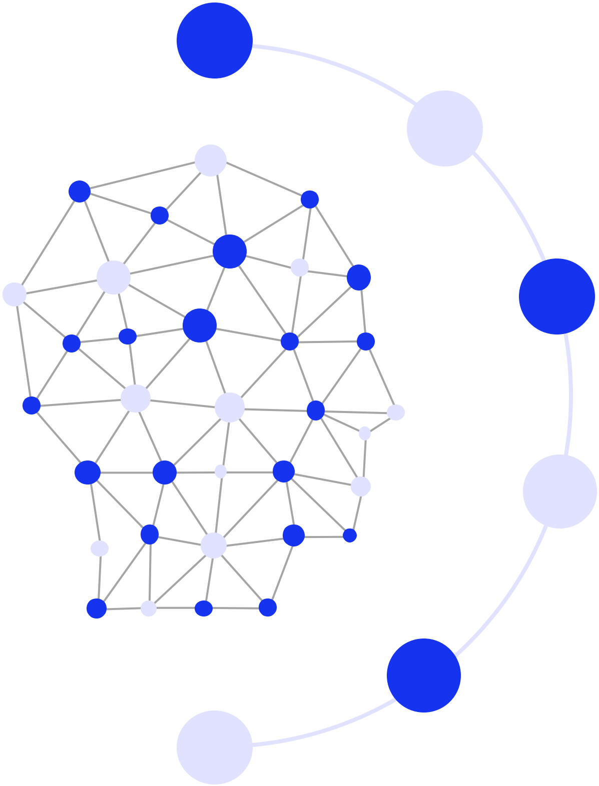
GenAI Optimization Techniques Part II

Optimization

Part 2

Chapter 3

Overview of Generative AI Principles



Generative AI

AI that creates new content—such as text, code, images, or music—rather than just analyzing data.

Neural Networks

Layers of connected artificial neurons that process data and learn complex patterns in large sets of data through training.

Large Language Models

Very large neural networks trained on massive text datasets to generate human-like language.

Transformer Architecture

The neural network design that powers modern LLMs using self-attention to understand word relationships.

Natural Language Processing

The field of enabling computers to understand, interpret, and generate human language..

Tokenization

The process of breaking text into small units (tokens) that an LLM can understand and process.,

Optimization Techniques

01 Fine-tuning

Training an LLM on custom data to specialize its behavior.

02 Prompt Engineering

Designing effective prompts to guide model outputs.

03 Retrieval-Augmented Generation (RAG)

Grounds LLMs with external knowledge sources.

04 Agentic AI

Orchestrates LLMs as multi-step, tool-using agents with memory and reasoning,

Cost

The amount of resource (data, compute, and engineering effort) needed to implement and maintain each technique.

Implementation Efficiency

How quickly and easily the technique can be deployed or iterated on in real-world workflows.

Performance

The degree of improvement the technique delivers in output quality, accuracy, and reliability.

Optimization Techniques

Technique	Description	Cost	Implementation
Fine-tuning	Further training enhances model performance.	💸💸💸💸	⚡⚡⚡⚡
Retrieval Augmented Generation (RAG)	Connects model to external data sources.	💸💸	⚡⚡
Prompt Engineering	Refines questions to maximize response quality.	💸	⚡
Agentic AI	Intelligent agents automate decisions and tasks.	💸💸💸	⚡⚡⚡

Part 2

Retrieval-Augmented Generation:

Demonstrates how to inject recent or domain-specific knowledge into models without altering their weights, improving accuracy.

Agentic AI Orchestration

Explains how to extend LLMs with reasoning, memory, and tools, enabling multi-step workflows and complex problem-solving.

RAG (Retrieval Augmented Generation)

High-level overview of RAG process for LLMs.

01 Retrieve Relevant Knowledge

- The system searches external sources (databases, documents, APIs) for context based on the user’s query
- Uses a vector database and embeddings to find semantically similar content

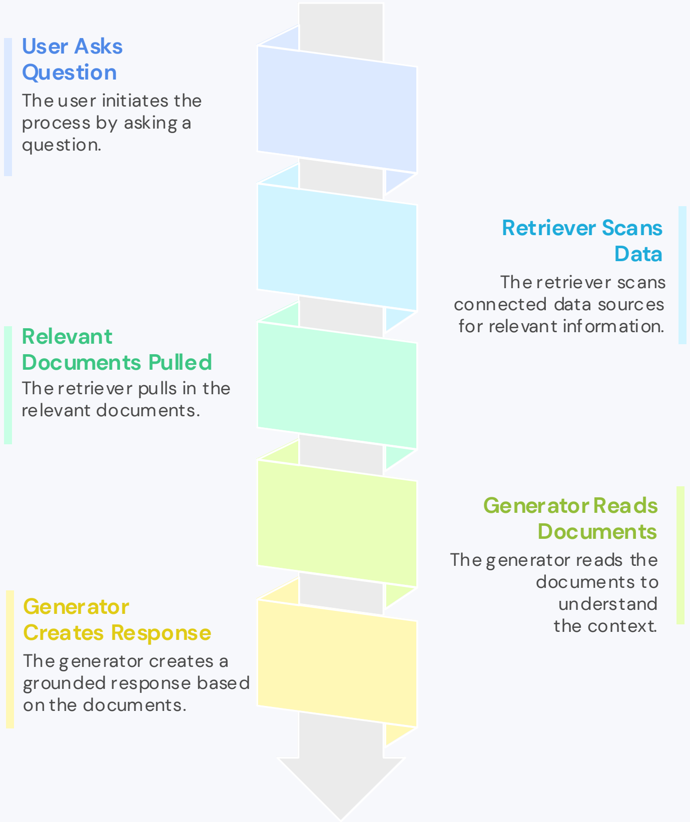
02 Augment the Prompt with Retrieved Context

- Injects the retrieved passages into the model’s prompt
- Gives the model up-to-date and domain-specific information before generating

03 Generate Answer Using the LLM

- The model produces an output that blends its own language skills with the retrieved facts
- This improves factual accuracy and reduces hallucinations

RAG Pipeline Flow



User Asks Question
The user initiates the process by asking a question.

Retriever Scans Data
The retriever scans connected data sources for relevant information.

Relevant Documents Pulled
The retriever pulls in the relevant documents.

Generator Reads Documents
The generator reads the documents to understand the context.

Generator Creates Response
The generator creates a grounded response based on the documents.

RAG – Analysis

Benefits	RAG enhances accuracy and freshness by giving models real-time access to external knowledge without retraining.
Drawbacks	It adds infrastructure complexity and depends on the quality of the retrieval data and indexing, which can affect speed and accuracy.

Sample Use Cases

Customer Support Bots

Providing real-time answers from product manuals and FAQs to reduce hallucinations.

Enterprise Knowledge Search

Answering employee questions from internal documents without fine-tuning the model.

Medical/Legal Q&A

Surfacing vetted references from trusted databases to keep outputs accurate and cite-able.

Optimization Techniques II

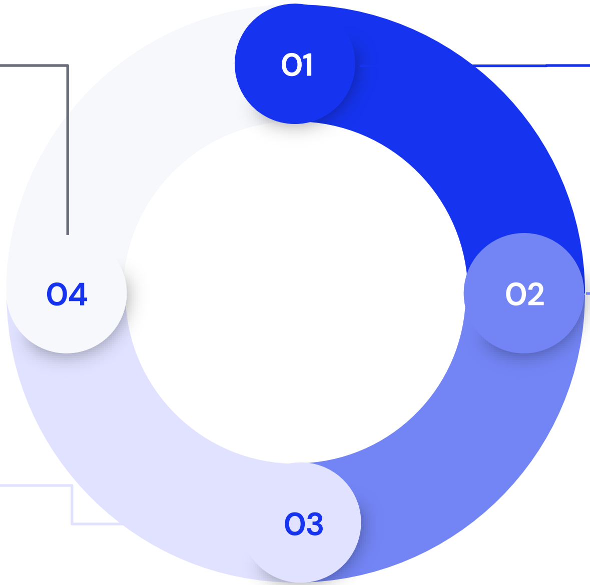
Optimizing LLMs with RAG

Result Delivery

The final product is shared, providing insightful, accurate, and data-rich responses or content.

Context-Based Generation

The model generates tailored outputs by leveraging both the integrated data and its inherent capabilities.



Data Retrieval

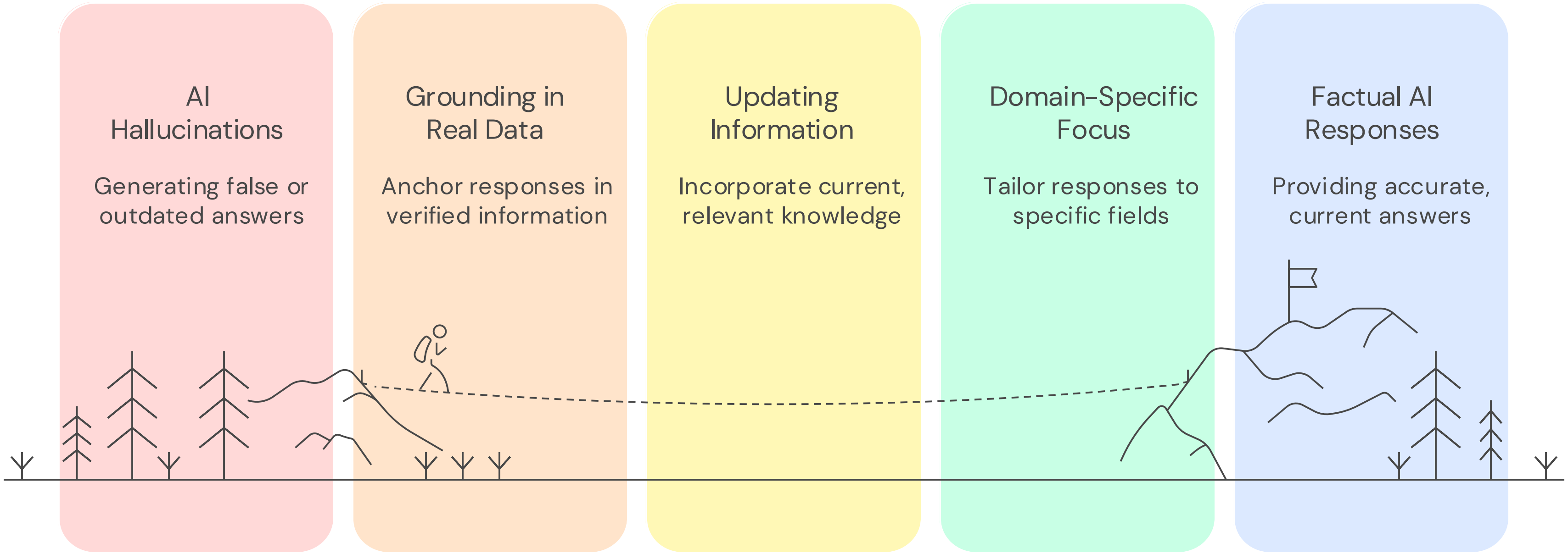
Using advanced retrieval methods, relevant real-world data is identified and extracted from diverse sources.

Information Integration

The retrieved data is processed and integrated into the LLM's framework, augmenting its knowledge and grounding its suggestions.

Optimization Techniques II

Reliable Answers with RAG



AI Hallucinations

Generating false or outdated answers

Grounding in Real Data

Anchor responses in verified information

Updating Information

Incorporate current, relevant knowledge

Domain-Specific Focus

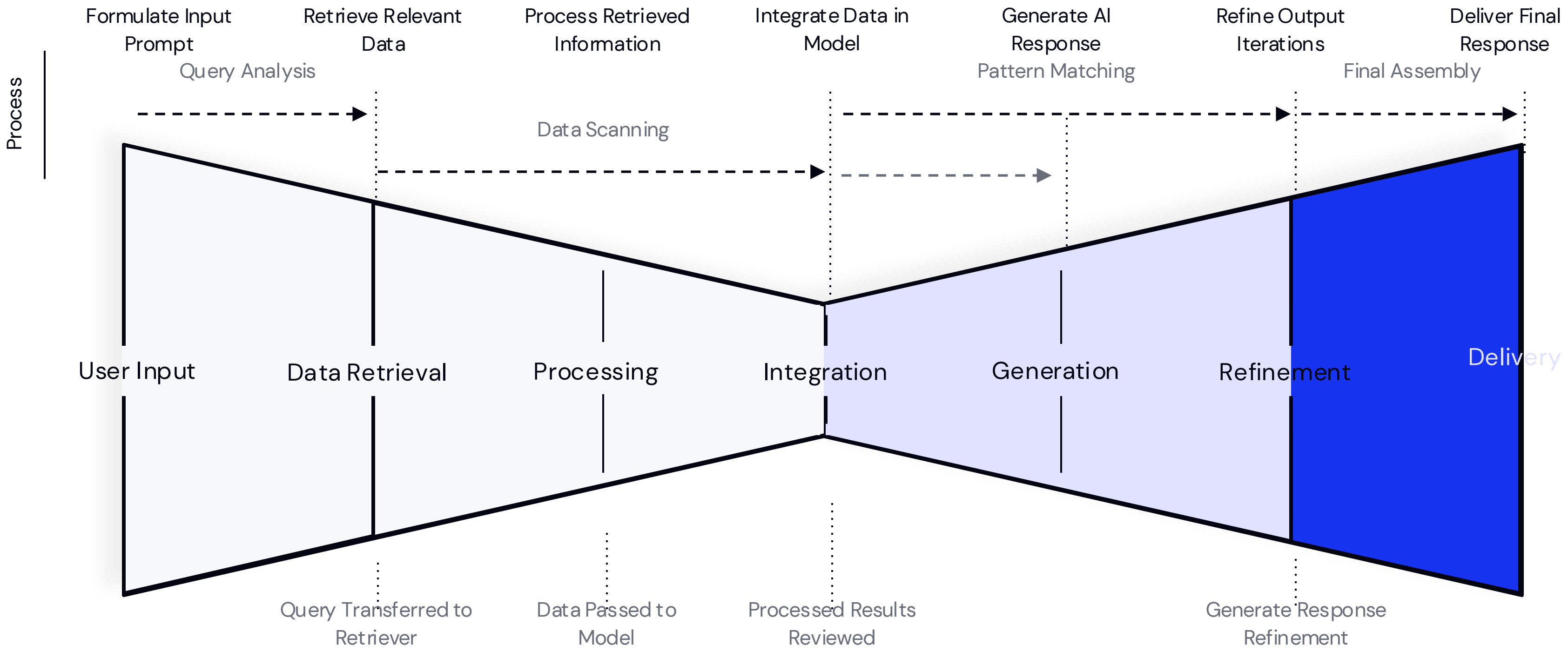
Tailor responses to specific fields

Factual AI Responses

Providing accurate, current answers

Optimization Techniques II

RAG Pipeline Visual Guide



Agentic AI Orchestration

High-level overview of Agentic AI Orchestration for LLMs.

01 **Implement an Orchestration Framework**

- Wrap the base LLM with a control layer (E.g., CrewAI, LangGraph, AutoGen, LlamaIndex, etc.)
- Enables planning, task decomposition, and coordination of multi-step workflows

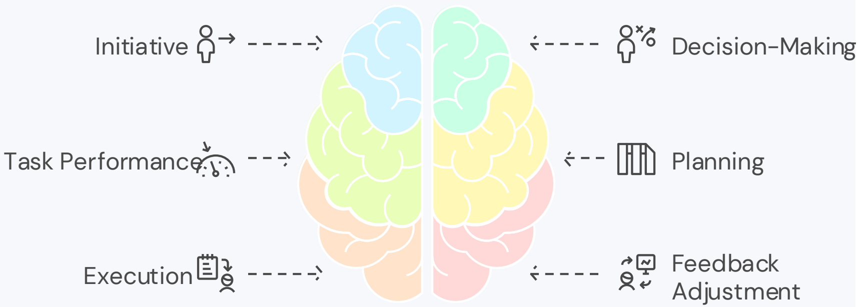
02 **Integrate Tools, Memory, and Context**

- Connect the agent(s) to external tools (APIs, code execution, retrieval systems)
- Provide memory (short- and long-term) so it can carry context across steps

03 **Implement Reasoning & Safety Loops**

- Add logic for reflection, evaluation, and self-correction
- Enforce guardrails, moderation, and approval gates to control behavior and risks

Agentic AI Core Components



Agentic AI – Analysis

Benefits

Agentic AI lets LLMs handle complex, multi-step tasks autonomously, greatly boosting performance and capability without changing their core weights.

Drawbacks

It adds architectural complexity, increases latency and cost, and requires strong safety mechanisms to prevent errors from compounding.

Sample Use Cases

Research Agents

Orchestrate multi-step searching, reading, and summarizing across sources.

Workflow Automation

Chain tools and memory to automate structured business processes.

Coding Agents

Combine planning, tool use, and feedback loops to iteratively write and debug code.

Key Traits of Agentic AI

Goal-Oriented

Understands objectives.

Autonomous

Executes tasks without input.

Reasoning

Breaks down and prioritizes.

Self-Reflective

Continuously improves results.

Agentic AI systems prioritize objectives, autonomously execute plans, adapt reasoning to dynamic tasks, and self-reflect to optimize their outputs and actions.

Optimization Techniques II

Key Insights into AI Orchestration Frameworks

Understanding AI Orchestration

AI orchestration frameworks coordinate how multiple models, tools, and data sources work together to complete complex tasks.

Framework Components

They typically include planning logic, memory modules, tool connectors, and control flows that manage task execution.

Integration Strategies

Effective orchestration depends on clean APIs, modular design, and context-passing mechanisms to seamlessly connect systems.

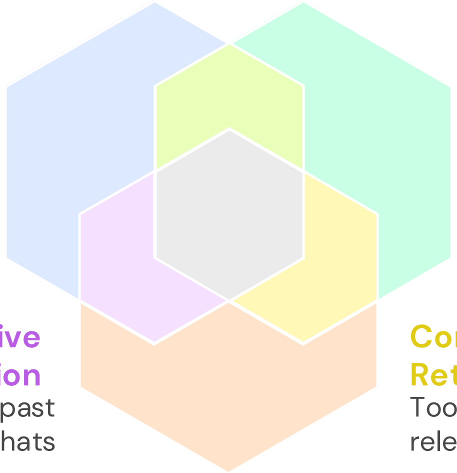
Implementation Considerations

For successful implementation of AI orchestration frameworks, ensure they are scalable, flexible, and integrate seamlessly into existing systems.

Prioritize sustainability and adaptability to dynamic changes in AI technology.

Agentic Orchestration Frameworks

How to turn an LLM into a reasoning Agent



Intelligent Tool Use
AI using tools for tasks

Language Models
Core AI reasoning engines

External Tools
Real-world API and data access

Adaptive Conversation
AI remembering past chats

Contextual Data Retrieval
Tools using memory for relevance

Context Memory
Retaining past interactions

Optimization Techniques II

From Prompt to Action Plan

Define objectives for the prompt.

Identify tasks for execution.

Create workflows for tasks.

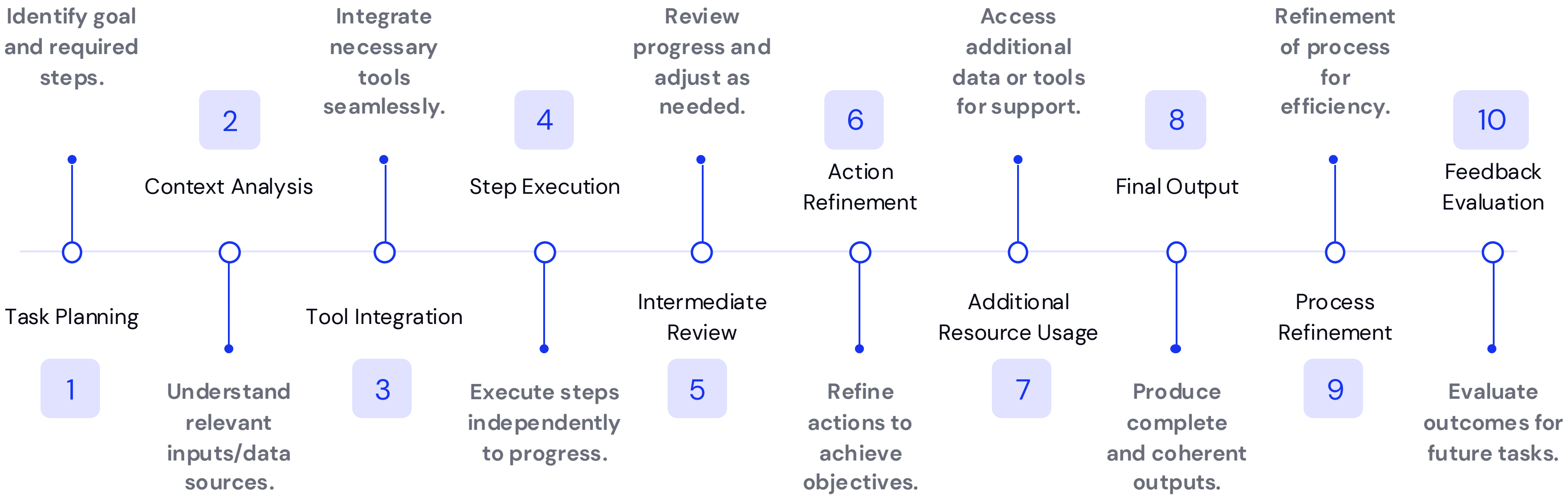
Allocate resources effectively.

Execute tasks as planned.

Review and adapt performance.

Agentic AI Task Execution

Autonomous Task Execution



1 Task Planning
Identify goal and required steps.

2 Context Analysis
Understand relevant inputs/data sources.

3 Tool Integration
Integrate necessary tools seamlessly.

4 Step Execution
Execute steps independently to progress.

5 Intermediate Review
Review progress and adjust as needed.

6 Action Refinement
Refine actions to achieve objectives.

7 Additional Resource Usage
Access additional data or tools for support.

8 Final Output
Produce complete and coherent outputs.

9 Process Refinement
Refinement of process for efficiency.

10 Feedback Evaluation
Evaluate outcomes for future tasks.

From LLM to Agent: The Agentic AI Design Canvas

Agent Roles

Define the roles or personas your agents will take on (researcher, planner, coder, analyst, assistant).

Core Capabilities

List what your agentic system can actually do (multi-step reasoning, tool use, retrieval, planning, autonomous task execution).

Feedback & Alignment

Describe how agents receive feedback, refine behavior, and stay aligned with goals and policies.

Model & Data Stack

Outline the components your agents rely on: LLM backbone, vector DBs, APIs, memory stores, orchestration framework, compute resources.

Interfaces & Surfaces

Specify how users will access the agents (chat UI, voice, web app, IDE plugin, Slack bot, API).

Agentic Behaviors

Map the core actions the system performs: planning, reasoning, task decomposition, tool invocation, error recovery, self-reflection.

Integrations & Dependencies

List external systems and services the agents must connect to: APIs, databases, CRMs, code execution environments, monitoring tools.

Metrics & Outcomes

Define success measures: task success rate, latency, user satisfaction, cost per run, autonomy level.

Constraints & Limits

Note system constraints and costs: token usage, API call limits, latency budgets, safety guardrails, governance policies.

Agentic AI Challenges

Current Challenges

1 **Unclear purpose:** It's hard to see where agentic AI fits into daily work.

2 **Complex systems:** Multi-step agents feel opaque and hard to trust.

3 **Limited readiness:** Teams lack skills and processes to support agent-driven work.

Negative Impacts

- **Slow adoption:** Humans hesitate to use tools they don't fully understand.
- **Frequent errors:** Poorly configured agents produce unreliable results.
- **Workflow friction:** Agents don't yet fit smoothly into existing systems.

- **User skepticism:** Confusion and mistakes reduce confidence in the tech.
- **Operational drag:** Teams spend more time managing the tools than benefiting from them.
- **Missed opportunities:** Early-stage failures make leaders hesitant to invest further.

Agentic AI Benefits

Future State

1 **Smarter workflows:** Agents handle routine steps so teams can focus on critical work.

2 **Seamless tools:** Agents connect data, apps, and systems behind the scenes.

3 **Adaptive support:** Agents learn from feedback to get more accurate over time.

Positive Outcomes

- **Less busywork:** Humans spend less time on repetitive, manual tasks.
- **Faster progress:** Projects move quicker when agents automate prep work.
- **More confidence:** Reliable outputs build trust and reduce second-guessing.

✓ **Happier teams:** Less grind boosts morale and engagement.

✓ **Stronger performance:** Teams can deliver higher-quality results faster.

✓ **Space to innovate:** Freed-up time fuels creativity, relationship-building and new ideas.

Thank you!

Questions?