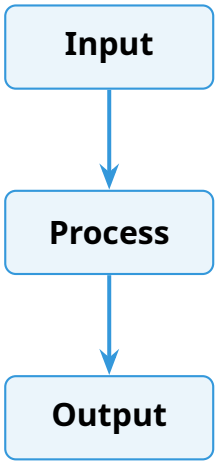
LangChain vs. LangGraph

Evolution of Structure: From Chain to Loop

LangChain (Building Blocks)

Feature: DAG (Directed Acyclic Graphs)

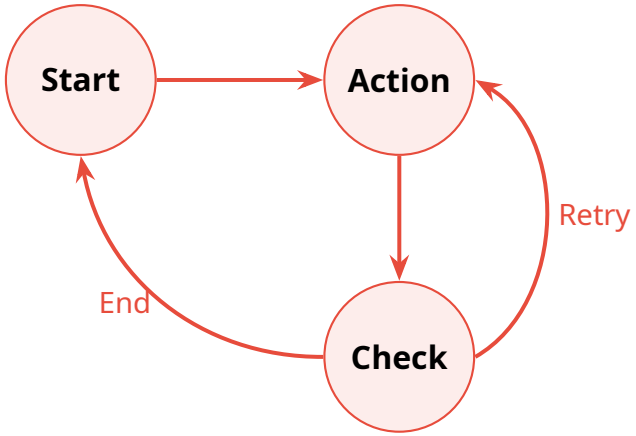
- **Unidirectional:** Flows only from start to end.
- **Linear:** Prompt → LLM → Output.
- **Components:** Prompts, Retrievers, Memory.
- **Limitation:** Difficult to implement complex looping logic.



LangGraph (Control Flow)

Feature: State Machine (Cyclic Graphs)

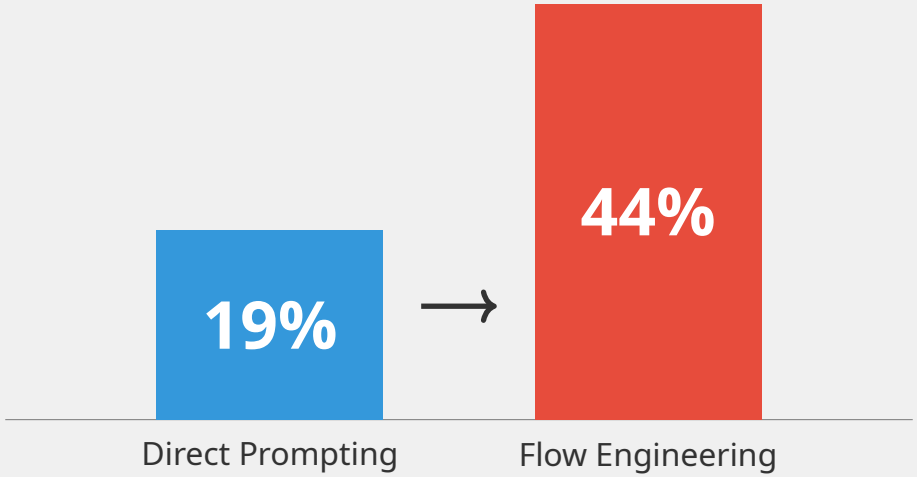
- **Cyclic:** Can return to previous steps.
- **State:** Memory shared across all steps.
- **Agentic:** Planning, Reflection, Self-Correction.
- **Core:** Nodes (Actions), Edges (Logic), Loops.



The Power of Flow Engineering

For complex tasks, **designing the flow** is more effective than prompt engineering.

AlphaCodium Research Results (GPT-4 Coding Accuracy):

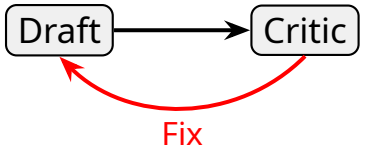


4 Key Patterns Using Graph Structures

1. Reflection & Self-Correction

Flow: Draft → Critic → (Error?) → Rewrite Loop

- Reduces hallucinations.
- Improves factual accuracy.



2. Decomposition

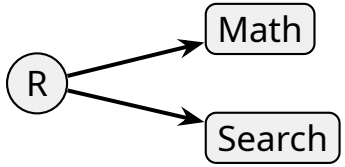
Flow: Planner breaks down steps → Sequential execution.

- Solves complex logic problems.
- Step-by-step reasoning.

3. Tool Use & Routing

Flow: Classify query (Router) → Specialized Agent.

- Increases relevance.
- Reduces noise.



4. Human-in-the-loop

Flow: Task → Pause (Approval) → Finalize.

- Ensures safety.
- Prevents critical errors.