DropTicks AI Article

Agent Frameworks in Practice: When to Use LangGraph, OpenAI Agents, or Simpler Code

AI Agents | Mar 09, 2026

A practical decision guide for choosing an agent runtime without overengineering your product.

Article

Agent frameworks are useful, but they are not automatically required. The right choice depends on workflow complexity.

Use simple application code when the flow is short: one model call, one retrieval step, one tool call, or a predictable form-to-output workflow. This keeps latency, debugging, and security easier.

Use an agent runtime when the workflow needs multiple tool calls, state, handoffs, resumable execution, guardrails, or human review. This is where frameworks such as LangGraph and OpenAI Agents become useful.

Use graph orchestration when the workflow has known stages and must be inspected or resumed. For example: classify request, retrieve context, call tool, verify output, ask for approval, write result.

Use agent handoffs when specialists should own different parts of a task. For example: research agent, coding agent, review agent, deployment agent.

The senior engineering rule is to make the control plane explicit. Do not hide business logic inside a prompt. The application should decide permissions, retries, logging, and stop conditions.

The most reliable agent systems look less like open-ended autonomous bots and more like constrained workflow engines with model-powered steps.

Web link
https://www.dropticks.com/articles/agent-frameworks-in-practice-when-to-use-langgraph-openai-agents-or-simpler-code

Related links
https://openai.github.io/openai-agents-python/
https://docs.langchain.com/oss/python/langgraph/overview
https://docs.haystack.deepset.ai/docs/intro