LangChain's Second Birthday

Reflections on how LangChain has evolved — including our products, ecosystem, and community — over the past two years, and where we're headed next.

HARRISON CHASE 6 MIN READ OCT 24, 2024

Today marks two years since the `langchain` Python package was released into the wild. Two years is both a long and short amount of time. On one hand, it still feels very early in our collective journey to build applications with LLMs. Yet. so much has also changed in that time. With the rapid pace of chang industry, we've mostly been heads-down building. But on the second

anniversary of our first release, we want to take stock and reflect on how `langchain` has evolved and where we're headed next.

New tooling, same mission: Enabling applications that can reason

The original tagline of `langchain` was to "connect LLMs to external sources of computation and data". This predated the term "agent", but that's essentially what the 1-liner described. Our mission from the beginning has been consistent: We want to make it as easy as possible to build context-aware, agentic applications.

However, what **has** changed is the tooling we offer to accomplish that mission. First and foremost, what started as `langchain` , the Python/JS open source library, has evolved into LangChain, the company! While `langchain` remains our most popular product, we've expanded with two additional key products — **LangGraph** and **LangSmith**.

In 2022, developers primarily needed an easy way to learn to build with this new technology and quickly get started. Today, the greater challenge is creating high-quality applications that work reliably and make a real impact. The evolution of all three of our products reflects this shift.

How the ecosystem has evolved

Two years ago, the whole generative AI ecosystem was incredibly nascent — ChatGPT hadn't even launched. `langchain` took off because it gave developers an easy entry point to build LLM apps quickly and keep up with the latest research. At that point, we had all the most popular LLM integrations (a

whopping 3!) and off-the-shelf chains that made it possible to get started with 5 lines of code and tinker with common use cases.

Since then, the ecosystem has evolved in three significant ways:

> The ecosystem has stabilized. While we still believe it is early days, the GenAI landscape is not nearly as nascent as it was two years ago. LLM applications are no longer being built solely in hackathons — they are being shipped to production.
>
> The ecosystem has exploded. When `langchain` first launched, there were approximately 3 LLM providers. Today, there are over 70 model providers, along with countless integrations (document loaders, vector stores, and more) - totaling over 700 different integrations in `langchain`.
>
> Most importantly, the ecosystem has matured. For enterprises and startups alike, the focus has shifted from simply building a prototype to making their LLM app production-ready.

Let’s dive into that last point. Our original open-source library `langchain` made (and still makes) it easy to get started with common LLM app use cases in just a few lines of code. But as the ecosystem has matured, it's become apparent that easy to get started != easy to get to production.

Naturally, this drove us to ask: what was preventing developers from shipping their apps to real users?

The need for LangSmith and LangGraph

LangSmith: Testing and observability for production-ready LLM apps

In our conversations with early `langchain` users and partners, we consistently heard about the unreliability of LLMs being a big roadblock. For most engineers, it wasn't enough to just integrate a model — getting the LLM to consistently make the right decisions took serious effort. Prompt engineering, in turn, evolved into its own discipline.

Recognizing this early on, we began developing **LangSmith** at the start of 2023 to address two key needs: **observability** and **evaluation**. Adding observability helps developers pinpoint where their LLM apps are messing up, while evaluation prevents regressions and ensures continuous app improvement. These two pillars — identifying LLM app issues and systematically improving performance — are still at the heart of LangSmith today.

As we built LangSmith, we worked closely with customers like Moody's, **Elastic**, **Podium**, and **Rakuten** to refine the product. Along the way, we continued gathering **best practices for testing & evaluating LLM apps** — and we saw how LangSmith helped AI/ML teams not only gain deep visibility into their AI assistant and chatbot interactions, but also take action to swiftly test and debug tricky issues.

LangGraph: Agent control with flexible orchestration

As we collaborated with talented builders, startups, and enterprises, another challenge emerged. By the early summer of 2023, it became clear that higher-level, prebuilt constructs in LangChain weren't enough for production use cases. We kept hearing that off-the-shelf chains were too rigid, and preconfigured agents lacked the reliability developers needed at scale.

We realized that in order to increase the reliability, developers **needed** more control over the cognitive architectures of their applications. Inspired by all the feedback, we built **LangGraph** — a flexible orchestration framework that lets developers build agents across a spectrum of autonomy, from precisely-defined workflows to more autonomous, adaptive agents. LangGraph provides a flexible architecture for users to customize their agents to their individual or business needs. With human-in-the-loop support and moderation loops, a LangGraph user has more control over their agent's behavior to ensure reliable outputs.

Already, we've seen exciting use cases built on LangGraph — from **Replit's coding agent** to **Unify's sales agent** — that have helped us validate the need for agents to reliably handle complex tasks.

LangChain open source today

So where does this leave the LangChain open source packages today, and what's changed from two years ago? Today, `langchain` is:

More stable. As the generative AI ecosystem has stabilized, so has `langchain`. We went to our first stable release in early 2024, and we've had only had two releases with breaking changes since.

More comprehensive. With the explosion of the GenAI, integrations continue to be a main part of `langchain`. Here, we are large beneficiaries of our incredible community. There are so many things to connect LLMs to — from vector stores, to retrievers, to document loaders — and we're immensely grateful for everyone who's contributed to our integration packages. We now also have **dedicated integration packages** for our top partners, and standardized tests for the various components.

More production-ready. As developers have shifted their focus from prototypes to production applications, we've adapted `langchain` accordingly. While building LangSmith to provide best-in-class observability and testing for LLM apps, we made sure it integrated seamlessly with `langchain`. And as we've built out the `langgraph` framework to enable powerful custom cognitive architectures, we've demonstrated how to integrate `langchain` components and migrate old `langchain` chains into more robust LangGraph applications.

As `langchain` has grown, our community has grown alongside us. This past year:

We've **doubled our contributors** in LangChain (across Python and JS) from 2000 to **4000**

We've **quadrupled the number of apps** powered by LangChain from 30k to **132k**.

Total downloads have jumped from 20 million to **130 million+** across Python and JS

A special shoutout to our LangChain community champions and ambassadors, who've helped us squash bugs, tackle issues, and have made educational content to make LangChain products accessible for everyone:

Aditya Thomas (sepiatone), Allen Firstenberg (afirstenberg), Andres Torres, Christophe Bornet (cbornet), Clemens Peters (clemenspeters), Colin McNamara (colinmcnamara), Daniel Nastase (js_craft_hq), Eden Marco (emarco177), Francisco Ingham (fpingham), gbaian10, Greg Kamradt (gregkamradt), Harry Zhang (zhanghaili0610), Marlene Mhangami (marlenezw), Rick Garcia (gitmaxd), Tom Spencer, Virat Singh (virattt)

The road ahead for LangChain

When I launched `langchain` two years ago, the goal was to make it as easy as possible to build LLM applications that reason. Now as a company, that same goal has motivated us over the past two years to improve `langchain` **and** to create orthogonal products that solve different needs.

We think this space is still in its early days and that there is a LOT of tooling to be built to make it enable developers to create game-changing applications. We think `langchain`, LangGraph, and LangSmith are key pieces of that puzzle — and we're confident that more components will surface over time. With the support and feedback from our incredible community, we'll continue to push LangChain forwards, exploring new directions while staying true to the vision of `langchain` that started it all.

Join the LangChain community

We'd love to have you along for the journey as we build the future of LLM applications — together! Here are some ways to get involved:

> Join the Slack community: **langchain.com/join-community**
>
> Contribute to the open source project: **python.langchain.com/docs/contributing/**
>
> Attend an event: **lu.ma/langchain**
>
> Get the latest product updates: **changelog.langchain.com**
>
> Learn more: **langchain.com/community**

TAGS

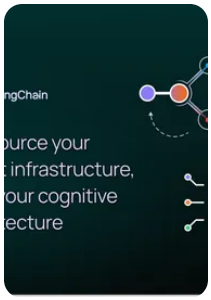
Harrison Chase

JOIN OUR NEWSLETTER

Updates from the LangChain team and community

Enter your email Subscribe

YOU MIGHT ALSO LIKE



Why you should outsource your agentic infrastructure, but own your cognitive architecture

HARRISON CHASE 3 MIN READ

What is a "cognitive architecture"?

HARRISON CHASE 3 MIN READ

What is an agent?

HARRISON CHASE 4 MIN READ

Sign up