# Building Generative AI Agents

## Using LangGraph, AutoGen, and CrewAI

Tom Taulli · Gaurav Deshmukh

Apress®

# Building Generative AI Agents

## Using LangGraph, AutoGen, and CrewAI

**Tom Taulli**
**Gaurav Deshmukh**

Apress®

*Building Generative AI Agents: Using LangGraph, AutoGen, and CrewAI*

Tom Taulli
Monrovia, CA, USA

Gaurav Deshmukh
Tarzana, CA, USA

ISBN-13 (pbk): 979-8-8688-1133-3
ISBN-13 (electronic): 979-8-8688-1134-0
https://doi.org/10.1007/979-8-8688-1134-0



Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Shiva Ramachandran
Development Editor: James Markham
Project Manager: Jessica Vakili

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a Delaware LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

If disposing of this product, please recycle the paper

# Table of Contents

# About the Authors

**Tom Taulli** is a consultant to various companies, such as Aisera, a venture-backed generative AI startup. He has written several books like *Artificial Intelligence Basics* and *Generative AI*. Tom has also taught IT courses for UCLA, PluralSight, and O'Reilly Media. For these, he has provided lessons in using Python to create deep learning and machine learning models. He has also taught on topics like NLP (natural language processing).

**Gaurav Deshmukh** is a highly skilled technology leader with over a decade of experience driving transformative software engineering initiatives. Throughout his career, he has held pivotal technical roles at prominent companies such as Guidewire, Cigna, Home Depot, American Agricultural Laboratory (AmAgLab), Tata Elxsi, and Amdocs. Gaurav's expertise encompasses a range of cutting-edge technologies, including cloud computing, cybersecurity, software automation, data engineering, and full-stack development with various programming languages and web technology frameworks. He employs his vast knowledge to create innovative solutions that optimize workflows and drive business growth. Gaurav holds both an MBA and a master's degree in Computer Science, with a focus on data warehousing and computer vision. He is dedicated to elevating the strategic role of software engineering in delivering business value.

## CHAPTER 1

# Introduction to AI Agents

Andrew Ng is a towering figure in the AI world. He has the rare blend of being an academic and entrepreneur.

When many in the tech world were focused on the dot-com boom during the 1990s, Ng saw AI as more interesting. While at Bell Labs, he worked on evaluating models, improving feature selection, and using reinforcement learning.

He would go on to get his master's degree in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology (MIT) and a Ph.D. in Computer Science from the University of California, Berkeley. His thesis was about reinforcement learning.

Ng would become a professor at Stanford. His course, which was CS229, was the most popular among students. He was also one of the first to see the usefulness of GPUs (Graphics Processing Units) for AI systems.

Ng would eventually apply his AI skills to the business world. He became the chief scientist at Baidu and helped to create Google Brain.

Then in 2011, he led the development of Stanford's MOOC (Massive Open Online Courses) platform. It would quickly attract large numbers of students.


T. Taulli and G. Deshmukh, *Building Generative AI Agents*,
https://doi.org/10.1007/979-8-8688-1134-0_1

Ng leveraged this experience by cofounding Coursera, which is one of the world's top online learning platforms. The company went public in 2021, with a market value of nearly $6 billion. Currently, it has about 148 million registered users and has partnerships with more than 325 universities and companies.[1]

After this, Ng founded other companies like DeepLearning.AI and Landing AI. He even has launched a venture capital fund.

No doubt, Ng has a knack for understanding trends—especially in the field of AI. This is someone who you should not bet against.

Then what is he looking at next? Where does he see the biggest opportunities?

It's with AI agents. He has noted that they are an "exciting trend" and something you "should pay attention to."[2] He has also said:

> *AGI (Artificial General Intelligence) feels like a journey rather than a destination. But I think … agent workflows could help us take a small step forward on this very long journey.*[3]

Ng is far from an outlier. Many of tech's most influential people are optimistic about AI agents.

Just look at Bill Gates. In his blog, he wrote:

> *In the computing industry, we talk about platforms—the technologies that apps and services are built on. Android, iOS, and Windows are all platforms. Agents will be the next platform.*

In his post, he details how software has changed little since he started Microsoft during the mid-1970s. The applications are "pretty dumb."

---

[1] https://investor.coursera.com/overview/default.aspx

[2] https://www.youtube.com/watch?v=sal78ACtGTc&t=125s

[3] https://www.youtube.com/watch?v=sal78ACtGTc&t=125s

But AI agents will change everything. A key part of this will be due to a system's understanding of your "work, personal life, interests, and relationships." In other words, software will become very smart—and much more useful and productive.

According to Gates:

> *Imagine that you want to plan a trip. A travel bot will identify hotels that fit your budget. An agent will know what time of year you'll be traveling and, based on its knowledge about whether you always try a new destination or like to return to the same place repeatedly, it will be able to suggest locations. When asked, it will recommend things to do based on your interests and propensity for adventure, and it will book reservations at the types of restaurants you would enjoy. If you want this kind of deeply personalized planning today, you need to pay a travel agent and spend time telling them what you want.*[4]

Then there is this take from McKinsey, which is one of the leaders in helping companies leverage AI technologies:

> *The value that agents can unlock comes from their potential to automate a long tail of complex use cases characterized by highly variable inputs and outputs—use cases that have historically been difficult to address in a cost- or time-efficient manner. Something as simple as a business trip, for example, can involve numerous possible itineraries encompassing different airlines and flights, not to mention hotel rewards programs, restaurant reservations, and off-hours activities, all of which must be handled across different online platforms. While there have been efforts to automate parts of this process,*

[4] https://www.gatesnotes.com/AI-agents

> *much of it still must be done manually. This is in large part because the wide variation in potential inputs and outputs makes the process too complicated, costly, or time-intensive to automate.*[5]

---

**Note** Sonya Huang is a partner at Sequoia Capital. She has backed some of the hottest generative AI startups like Hugging Face, Glean, and LangChain.[6] According to her: "One of our core beliefs is that agents are the next big wave of AI, and that we're moving as an industry from copilots to agents."[7]

---

# What Are AI Agents?

There is no clear-cut definition of AI agents. But this should come as no surprise. The category for AI agents is still in the nascent stages—and the technology is moving quickly. Just as the Internet grew to encompass a vast array of applications and services, AI agents are likely to undergo a similar trajectory of rapid development and diversification. This means that developers are at a point where significant opportunities for growth and excitement abound.

Yet we still need a basic definition. So what should this be? A good place to start is with one of the pioneers of the generative AI revolution, Harrison Chase. He is the cofounder of LangChain, which is one of the most popular development frameworks for this technology.

---

[5] https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/why-agents-are-the-next-frontier-of-generative-ai

[6] https://www.linkedin.com/in/sonyaruihuang/details/experience/

[7] https://www.sequoiacap.com/podcast/training-data-harrison-chase/

Here's how he defines generative AI agents:

> *The way that I think about agents is that it's when an LLM is kind of like deciding the control flow of an application. So what I mean by that is if you have a more traditional kind of like RAG chain, or retrieval augmented generation chain, the steps are generally known ahead of time, first, you're going to maybe generate a search query, then you're going to retrieve some documents, then you're going to generate an answer. And you're going to return that to a user. And it's a very fixed sequence of events.*[8]
>
> *And I think when I think about things that start to get agentic, it's when you put an LLM at the center of it and let it decide what exactly it's going to do. So maybe sometimes it will look up a search query. Other times, it might not, it might just respond directly to the user. Maybe it will look up a search query, get the results, look up another search query, look up two more search queries and then respond. And so you kind of have the LLM deciding the control flow.*

Another way to look at AI agents is to understand their components. They include reflection, tools, memory, planning, multi-agent collaboration, and autonomy.

Let's take a look at each.

## Reflection

Reflection in AI agents refers to the ability of a system to inspect and adjust its own cognitive processes. This self-awareness allows the AI to scrutinize its decision-making, learning patterns, and problem-solving approaches. By engaging in reflection, AI can break down intricate challenges, extract insights from its experiences, and offer clearer justifications for its conclusions.

[8] https://www.sequoiacap.com/podcast/training-data-harrison-chase/

Recent research, such as the Reflexion framework, has demonstrated the significance of self-reflection in enhancing AI capabilities. Reflexion uses verbal self-reflections to generate valuable feedback for future trials, storing this feedback in the agent's memory. This process involves an iterative optimization where the agent evaluates its actions, receives feedback, and adjusts its behavior accordingly. This method has shown improvements in tasks like decision-making, reasoning, and programming.

This metacognitive ability enhances AI systems' flexibility and resilience. As the AI evaluates its past performance and outcomes, it can refine its strategies and expand its autonomous capabilities. The process facilitates error detection, strategic evolution, and more efficient goal attainment. For example, Reflexion agents have demonstrated improved performance in environments such as AlfWorld and on tasks like search-based question answering and code generation.[9]

## Tools

Tool use in generative AI agents refers to their ability to interact with external tools, APIs, or software to enhance their capabilities and perform complex tasks. This feature allows AI systems to go beyond their core functions like language or image generation. AI agents can access up-to-date information, retrieve real-time data, perform calculations, manipulate files, and automate workflows by chaining multiple actions together. This integration significantly improves their accuracy and expands their domain knowledge.

Examples of tool use in generative AI include web browsing for current information, code execution in real-time environments, data analysis and visualization, calendar management, file operations, and complex

[9] https://ar5iv.labs.arxiv.org/html/2303.11366

mathematical computations. These capabilities enable AI agents to handle a broader range of tasks effectively. For instance, Salesforce's Einstein GPT integrates with CRM tools to provide AI-generated content across various business functions. Similarly, AWS's Solution Architect Agent uses custom-built tools to query AWS documentation, generate code, and create architectural diagrams.

Complex tasks often occur in dynamic environments where solutions are not immediately apparent or may change unexpectedly. For instance, data sources might be unavailable, requiring the search of alternative sources, or actions might have unforeseen side effects. In app development, an initial API request might fail due to network issues, incorrect argument formats, or changes to the API itself. To adapt, an agent might need to retry the request with different parameters based on feedback, such as error messages, or seek explicit human assistance.

What if there is no API? A successful agent must be capable of navigating an end-user interface. This task is complex, requiring the agent to understand the interface content, whether by processing HTML elements or interpreting pixels in a screenshot. The agent then needs to determine the appropriate action, such as clicking a button or filling out a form, and verify the success of this action by checking for confirmation messages. Each action alters the interface state, influencing subsequent actions and requiring the agent to continuously adapt its approach.

## Memory

Memory in AI agents is a critical capability that enables these systems to retain and utilize information from previous interactions or tasks. This function allows AI to maintain context, learn from experiences, and deliver more coherent and personalized responses.

There are different variations. First, there is short-term memory. It temporarily retains and manipulates information relevant to the immediate task. It tracks recent events or data points needed briefly before

being discarded or transferred to long-term memory. Implementation often involves maintaining a log of recent actions or the last few conversational turns.

Then there is long-term memory. At a high level, this type provides the agent with the ability to retain and access information over extended periods, storing accumulated knowledge, learned experiences, and established patterns. It shapes the agent's decision-making processes and adaptability.

Long-term memory is often implemented using vector databases. This allows for efficient retrieval of relevant information based on queries related to events, descriptions, and associated metadata. The structure, representation, and retrieval mechanisms of this data significantly impact the effectiveness of memory recall and the overall performance of the AI agent.

Long-term memory includes

- Episodic Memory: This stores specific events or experiences, allowing the agent to recall past occurrences and apply learned lessons to current situations.
- Semantic Memory: This retains general knowledge and facts about the world, enabling the agent to understand objects, concepts, relationships, and procedures. It provides a broad understanding of the domain, allowing reasoning and inference even in unfamiliar scenarios.
- Procedural Memory: This focuses on storing learned skills and procedures, emphasizing how to perform tasks rather than recalling specific events.

Recent research highlights the effectiveness of these memory systems. For instance, a study demonstrated how AI agents with short-term, episodic, and semantic memory systems outperformed those without such structured memory in complex environments. This highlights the benefits of these memory types for task performance and learning efficiency (Kim et al., 2023).[10]

The development of advanced memory systems in AI agents is crucial for their ability to handle increasingly sophisticated tasks with greater independence. For example, the JARVIS-1 agent uses multimodal memory to enhance its task planning and execution in complex, open-world environments, demonstrating significant advantages in performance and adaptability (Weng, 2023).[11]

## Planning

Planning in AI agents involves leveraging LLMs to autonomously determine a sequence of steps necessary to achieve a broader objective. This process allows AI to break down complex goals into manageable tasks. This enhances its capability to execute intricate projects. For example, an LLM can guide an AI agent in organizing a virtual event by breaking the task into smaller steps such as selecting speakers, scheduling sessions, and coordinating technical support.

Recent advancements illustrate the profound impact of LLM-based planning on autonomous agents. The Reflexion framework, for instance, combines planning, self-reflection, and memory to iteratively enhance task performance. This allows agents to dynamically adjust their plans based on feedback and previous experiences. This helps to improve decision-making and execution over time.

[10] https://ojs.aaai.org/index.php/AAAI/article/view/25075
[11] https://ar5iv.labs.arxiv.org/html/2311.05997

Furthermore, the TPTU (Task Planning and Tool Usage) framework emphasizes the synergy between planning and tool usage. This framework evaluates how effectively LLMs can plan tasks and use tools. AI agents can either adopt a one-step approach, which outlines the entire task at once, or a sequential approach, which addresses each subtask individually, allowing for ongoing feedback and adjustments.

In practical scenarios, planning enables AI agents to manage tasks that require dynamic responses and specialized knowledge. For example, an AI agent tasked with automating a home garden can plan steps such as setting up sensors, configuring irrigation schedules, monitoring plant health, and integrating data with a smartphone app.

While planning significantly enhances AI capabilities, it also introduces unpredictability, as agents might deviate from expected behaviors due to the complexity of generating dynamic plans. However, with ongoing advancements in this field, the reliability and sophistication of planning in AI agents are anticipated to improve.

## Multi-agent Collaboration

Multi-agent collaboration uses various LLMs that work together to accomplish complex tasks. This approach is similar to how human teams operate—that is, each agent specializing in different subtasks to achieve a common goal. For example, in a marketing campaign project, different AI agents could assume roles such as content creator, market analyst, campaign strategist, and performance evaluator.

By prompting one or multiple LLMs to perform distinct tasks, you can create specialized agents. For instance, in a marketing campaign, an agent tasked with content creation might be prompted with instructions like, "You are an expert in crafting engaging marketing copy. Write content for the campaign focused on promoting the new product…." This method leverages the strengths of LLMs while maintaining a clear focus on specific subtasks, enhancing overall performance and efficiency.

Another agent could be assigned to market analysis with a prompt such as, "You are skilled in analyzing market trends and consumer behavior. Provide insights based on the latest data to inform the campaign strategy."

Research has shown that multi-agent systems often outperform single-agent setups. Studies like those from MIT demonstrate that collaborative interactions among multiple AI models can significantly improve reasoning and factual accuracy.[12] By engaging in deliberative processes, these agents can critique each other's outputs, leading to more accurate and comprehensive solutions.

## Autonomy

AI agents exhibit autonomy by independently making decisions and executing tasks without constant human intervention. This autonomy stems from their ability to process data, learn from experiences, and adapt to new situations in real time. Advanced algorithms and machine learning techniques enable these agents to evaluate their environments, recognize patterns, and predict outcomes, allowing them to take actions that align with their programmed goals. For instance, in autonomous vehicles, AI agents must constantly interpret sensor data to navigate roads, avoid obstacles, and make driving decisions that ensure safety and efficiency. These decisions are made on the fly, showcasing the agents' ability to function autonomously in dynamic environments.

Moreover, AI agents enhance their autonomy through continuous learning and adaptation. Machine learning models allow agents to learn from their experiences and improve their performance over time. This

[12] https://news.mit.edu/2023/multi-ai-collaboration-helps-reasoning-factual-accuracy-language-models-0918

learning process involves analyzing past actions and outcomes to refine future strategies. For example, in customer service applications, AI agents can learn from previous interactions to provide more accurate and personalized responses in subsequent engagements.

However, it is often unwise to have a completely autonomous AI agent. Instead, there is a spectrum of autonomy and control that should be considered. Human oversight remains crucial in many scenarios to ensure that AI agents' actions align with broader ethical standards, safety protocols, and organizational goals. By balancing autonomy with human control, we can leverage the strengths of AI while mitigating risks associated with unsupervised decision-making.

Yes, there is much that goes into an agent. But this does not imply that you need to use all the components. You may need only a couple. It depends on the use case.

## UI and UX

The user interface (UI) and user experience (UX) are crucial components of software applications. They directly impact user satisfaction, engagement, and productivity.

A well-designed UI ensures that the software is visually appealing and intuitive, making it easier for users to navigate and accomplish their tasks efficiently. Good UX design, on the other hand, focuses on the overall experience users have with the application, including ease of use, accessibility, and responsiveness. Together, UI and UX design help reduce the learning curve for new users, minimize errors, and enhance the overall effectiveness of the software.

This not only boosts user satisfaction but also drives higher adoption rates and customer loyalty. A study by Forrester Research found that a well-designed UI could increase a website's conversion rate by up to 200%, while better UX design could yield conversion rates up to 400%.[13]

As AI agents evolve, rethinking UI and UX design becomes essential to deal with the unique challenges posed by LLMs. Given that LLMs are not always perfect and can sometimes be unreliable, traditional chat interfaces have been an early approach. This interface allows users to easily see the AI's actions, receive streamed responses, correct the AI by responding to it, and ask follow-up questions. This interactive and transparent format ensures that users can remain in control and make necessary changes.

However, there are limitations to this approach. The human remains very much in the loop, making the system more of a copilot rather than an autonomous operator.

One way to address this balance is by ensuring transparency and accountability in the AI's actions. For instance, in a home automation scenario, having a detailed log of everything the agent has done allows users to review and modify actions if necessary.

This review process could be streamlined through an interface that lets users easily modify the schedule for devices like lights, thermostats, and security systems. The AI can autonomously manage these devices, but users can still step in to adjust settings or provide feedback, which the AI can then learn from and adapt to in future tasks.

Moreover, the interface for interacting with AI agents can be designed to be more proactive and integrated into everyday devices. Instead of requiring users to open an application, the AI could work in the background and periodically reach out with updates or queries. For example, an AI agent might notify you through your smart home hub or

[13] https://www.forrester.com/report/The-Business-Impact-Of-Customer-Experience-Q4-2016/RES137870

wearable device with a message like, “Your energy consumption is higher than usual today. Would you like me to adjust the thermostat settings to save energy?”

This proactive approach ensures that AI agents are seamlessly integrated into users’ lives, providing assistance as needed without requiring constant manual engagement.

Ultimately, rethinking UI and UX for AI agents involves creating systems that are both user-friendly and capable of operating with a degree of autonomy while maintaining transparency and reliability. This ensures that users can trust AI agents to handle tasks efficiently, intervening only when necessary to ensure the desired outcomes.

## New Approaches to Development

Traditional software development follows a fairly deterministic workflow. It is based on a structured and sequential approach to creating software applications. This process typically begins with requirement analysis, where the needs and objectives of the software are clearly defined. This is followed by system design, where the architecture and detailed specifications are created. Next comes implementation or coding, where developers write the actual code according to the design specifications. Once the coding is complete, the software undergoes rigorous testing to identify and fix any bugs or issues. After successful testing, the software is deployed into the production environment. Finally, maintenance and updates are performed as necessary to address any issues that arise after deployment.

The deterministic nature of traditional software development lies in its predictability and repeatability. Each phase of the development process is well-defined and follows a linear progression. The clear documentation and structured processes make it easier to manage large teams and complex projects.

Developing generative AI agents significantly differs from traditional software development due to its reliance on probabilistic outcomes rather than deterministic processes. This can be a major adjustment for developers.

Let's take a look at a typical workflow. The first step is to identify the use case, a task that can be complex since certain scenarios may not be suitable for AI due to the need for predictability. Once a suitable use case is determined, selecting one or more models is the next challenge. This selection process is intricate because models are sophisticated and frequently updated.

Cost is another critical factor in developing generative AI agents. Whether using an API or running models locally, the expenses can be substantial. Running a model locally may require buying costly hardware, such as GPUs. Furthermore, the complexity of the workflows must be thoroughly evaluated. Given that LLMs operate on probabilities, there is always the risk of incorrect outputs or decisions. To mitigate these risks, implementing guardrails and considering options for a human-in-the-loop are common practices to ensure safety and accuracy.

Testing generative AI agents presents its own set of challenges due to the unpredictability of the responses. This testing phase can be lengthy and detailed, requiring extensive trials to ensure reliability and effectiveness.

According to Sonya Huang and Pat Grady, who are partners at Sequoia Capital:

> *Existing monitoring tools don't provide the level of insights you need to trace what went wrong with an LLM call. And testing is different in a stochastic world, too—you're not running a simple "test that 2=2" unit test that a computer can easily verify. Testing becomes a more nuanced concept with*

> *techniques like pairwise comparisons (e.g. Langsmith, Lmsys) and tracking improvements/regressions. All of this calls for a new set of developer tools.*[14]

To improve accuracy, it is often necessary to use databases with proprietary information, adding another layer of complexity. This may involve fine-tuning the model or employing techniques like Retrieval-Augmented Generation (RAG) to enhance the model's performance. Each of these steps underscores the dynamic and adaptive nature of developing generative AI agents. This certainly highlights the differences from the more deterministic workflows of traditional software development.

# Flavors of AI Agents

AI agents come in two primary forms: embodied agents and software agents. Each type serves distinct purposes and operates in different environments. They leverage the unique capabilities of AI to address specific needs and challenges.

Embodied agents are AI systems that interact with the physical world or simulated 3D environments. These agents are often used in robotics, where they can perform tasks such as assembly line work, warehouse management, and autonomous navigation. In video games, embodied agents control non-player characters (NPCs), creating more immersive and realistic experiences for players. The development of embodied agents requires sophisticated algorithms that enable perception, decision-making, and action within dynamic environments. These agents often rely on sensors, cameras, and other input devices to gather information about their surroundings, process this data in real time, and execute appropriate actions.

[14] https://www.sequoiacap.com/article/goldilocks-agents/

Software agents, on the other hand, operate within digital environments, handling tasks related to office work, workflows, and data management. These agents can automate repetitive tasks, manage emails, schedule appointments, and facilitate complex business processes. Software agents are designed to improve productivity and streamline operations by acting as intelligent assistants that can understand and execute various commands based on user inputs.

The development of both embodied and software agents involves distinct challenges and methodologies. Embodied agents require extensive training in real or simulated environments to handle physical tasks effectively. This training often involves reinforcement learning, where agents learn through trial and error to optimize their actions. Conversely, software agents are typically trained on large datasets using LLMs to understand and generate humanlike responses.

As for this book, the primary focus will be on software agents.

## Brief History

AI agents have been around since the dawn of AI, with early programs in the 1950s laying the groundwork for their development. The Logic Theorist (1955), created by Allen Newell and Herbert A. Simon, was among the first AI programs, designed to mimic human problem-solving skills by proving mathematical theorems from *Principia Mathematica*. Its use of automated reasoning and heuristics showcased the potential for machines to perform intelligent tasks. Following this, Newell and Simon developed the General Problem Solver (1957), a more versatile system capable of applying general strategies to solve a wide range of problems. Introducing means-end analysis and hierarchical problem-solving, GPS aimed for universal applicability, influencing both AI and cognitive psychology. These foundational efforts demonstrated that machines could emulate human reasoning and inspired future AI advancements.