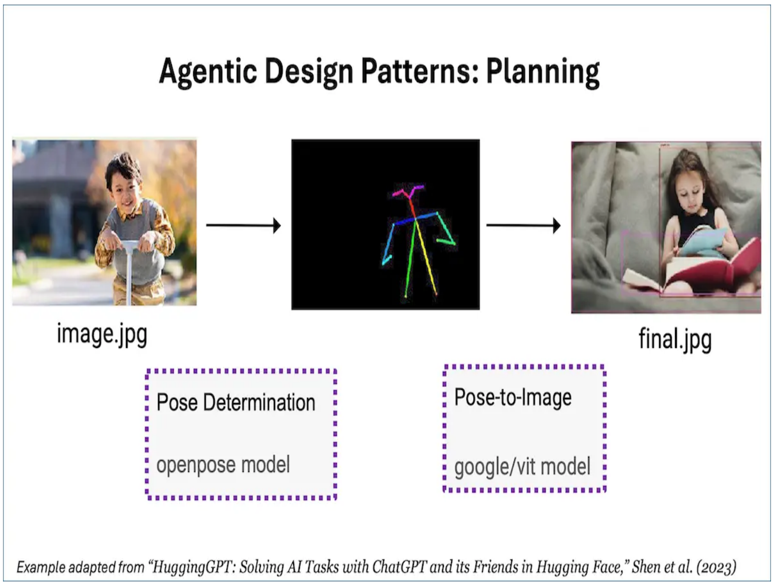
The transformative potential of AI is realized through the development and deployment of innovative applications. Generative AI significantly accelerates this process, enabling rapid prototyping and experimentation. This speed allows for the testing and refinement of multiple hypotheses, fostering

and refinement of multiple hypotheses, fostering a culture of innovation.

However, understanding the broader AI ecosystem is crucial. While focusing on building powerful models is essential, it's equally important to consider the underlying infrastructure, including hardware resources, data storage, and processing power. A holistic understanding of the AI stack,

power. A holistic understanding of the AI stack, encompassing both software and hardware layers, is necessary to fully leverage the potential of AI.

The shift towards agentic workflows represents a significant advancement in AI interaction, moving away from single-prompt, direct-response models. Agentic workflows involve multiple steps, including research, planning, execution, and refinement, mirroring human problem-solving. This iterative

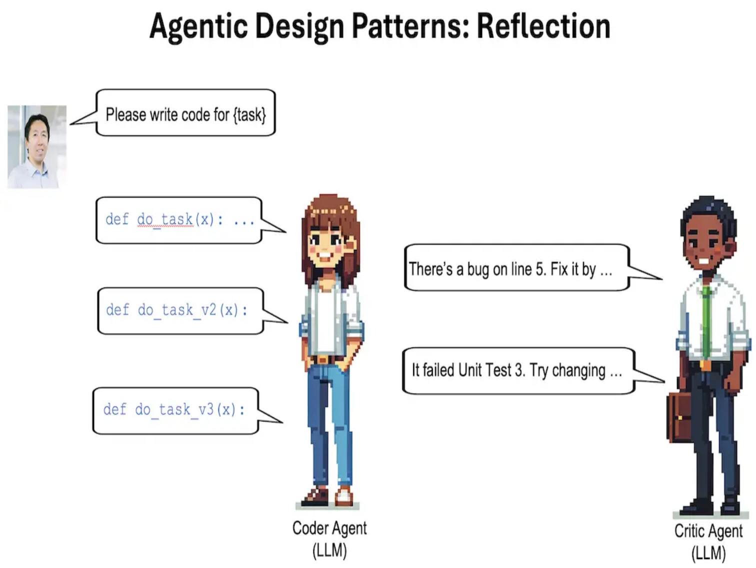


mirroring human problem-solving. This iterative approach often yields improved outcomes, particularly for complex tasks requiring nuanced reasoning.

While agentic workflows can incorporate LLMs, they are not limited to them. Components can come from various sources, including different AI models and data sources. The key principle is that these components collaborate to produce a better, more comprehensive response than a single LLM could

comprehensive response than a single LLM could achieve alone. For example, an autonomous vehicle might use an agentic workflow with agents for navigation, object detection, communication, and safety, each contributing to the overall goal of safe and efficient driving.

The Reflection design pattern empowers AI agents to evaluate and refine their own outputs, much like peer review in academia or debugging in software development. This process involves an AI agent analyzing its work to identify errors, inefficiencies, or areas for improvement. Through iterative



or areas for improvement. Through iterative cycles of self-assessment and revision, the agent can achieve more accurate and higher-quality results. This can be a single agent reflecting on its work or a multi-agent system where one agent generates content and another provides constructive

content and another provides constructive criticism.

AI agents can enhance their capabilities by leveraging external tools and APIs, known as the Tool Use design pattern. These tools can encompass information retrieval systems like web search engines, computational tools such as code interpreters, and interfaces for interacting with real-world

and interfaces for interacting with real-world applications. By integrating with these external resources, AI agents become more versatile and capable of tackling a wider range of complex tasks that require interaction with the external environment or specialized functionalities.

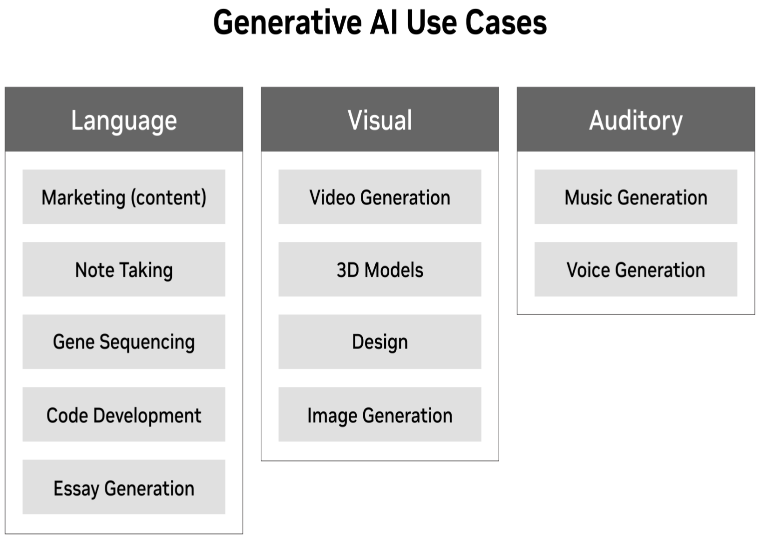
This decomposition of a complex task into smaller, manageable steps allows AI agents to approach problems in a structured and organized manner, similar to project management methodologies. By planning the necessary steps to achieve a complex objective, AI agents can better manage complexity and

AI agents can better manage complexity and increase the likelihood of successful task completion. This planning process can be static, with the entire plan defined upfront, or dynamic, where the agent adjusts its plan based on new information or unexpected events.

AI agents can be assigned specialized roles or functionalities within a multi-agent collaboration, allowing them to contribute their unique expertise to the task. These agents can communicate and interact, exchanging information and coordinating their efforts to solve complex problems

their efforts to solve complex problems collaboratively. This teamwork approach can lead to enhanced performance and the ability to tackle more intricate challenges.

Visual AI encompasses technologies enabling computers to interpret and understand visual information from images and videos. Its applications are vast, spanning autonomous vehicles, medical imaging analysis, quality control, and retail analytics. Effectively processing visual data alongside other



processing visual data alongside other unstructured data like audio and text is crucial for developing powerful and versatile AI systems. Agentic AI builds upon compound LLMs, strategically linking multiple LLMs in sequence to enhance task outcomes. This approach can involve an LLM for initial

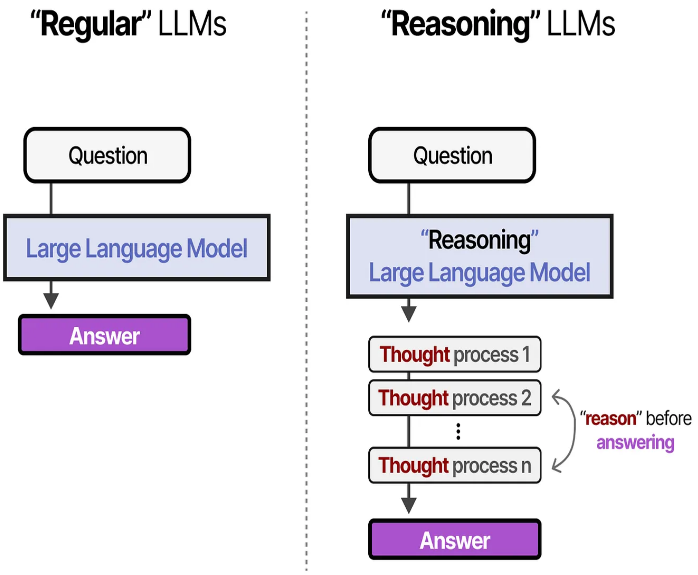
This approach can involve an LLM for initial drafting, another for critical review, and a third for revision, reflecting an iterative refinement process. Compound LLMs demonstrate the value of combining specialized AI functions, achieving higher quality and sophistication in output compared to

quality and sophistication in output compared to single LLMs.

Agentic AI expands on compound LLMs by utilizing a diverse set of agents, which can include LLMs, data retrieval systems, or APIs, each chosen for specific capabilities. An orchestrator agent manages the workflow, dynamically selecting and directing the appropriate agents based on the task's needs

the appropriate agents based on the task's needs and changing conditions. This allows for flexible and adaptable problem-solving, contrasting with rigid pre-defined workflows. The development of agentic AI builds upon the concept of compound LLMs, which involve linking multiple LLMs in sequence to

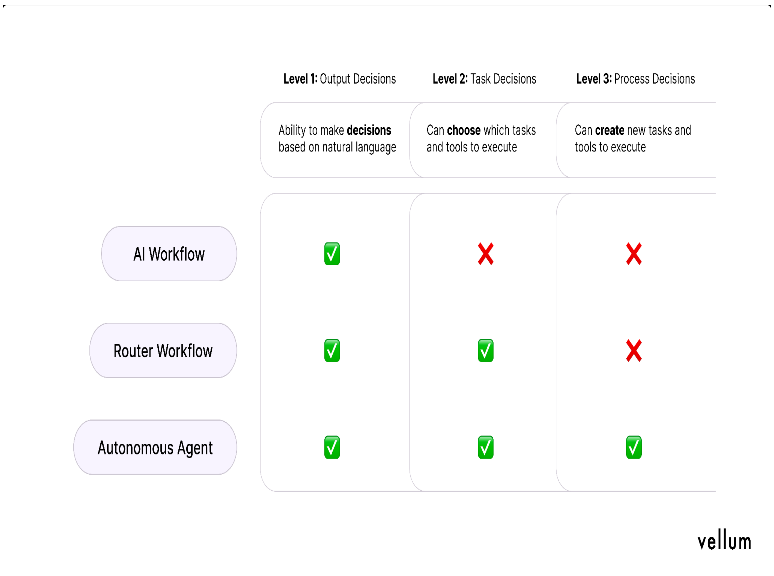
involve linking multiple LLMs in sequence to enhance task outcomes. This can be seen in a system using one LLM for drafting, another for critiquing, and a third for revision, showcasing an iterative refinement process.

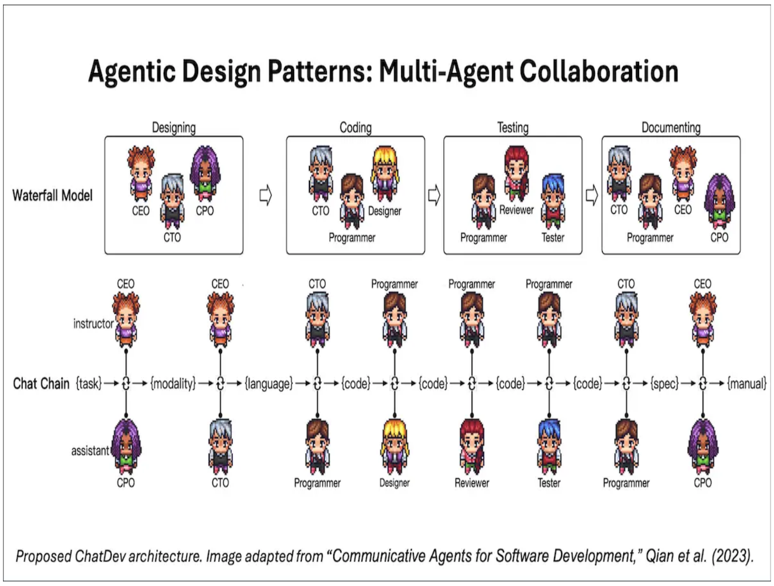


Agentic AI systems are designed to be adaptable and resilient in dynamic environments. This adaptability comes from their ability to modify their strategies based on real-time feedback or changing conditions. This makes them more effective in real-world situations compared to less flexible AI

situations compared to less flexible AI architectures that struggle to adjust to unforeseen circumstances.

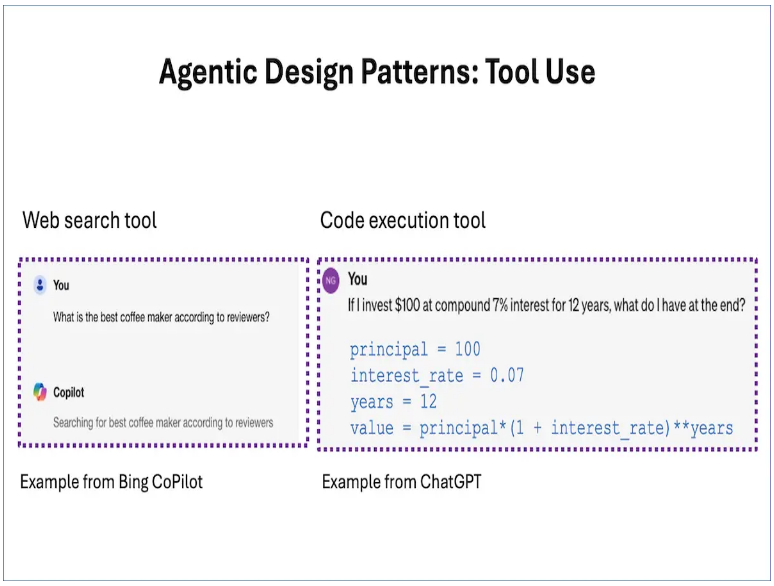
AutoGen is a framework that allows developers to build multi-agent systems. It supports conversational workflow orchestration, enabling agents to engage in looping dialogues until a task is completed. This is particularly useful for complex tasks like code generation, debugging, or multi-step





Proposed ChatDev architecture. Image adapted from "Communicative Agents for Software Development," Qian et al. (2023).

like code generation, debugging, or multi-step problem solving. AutoGen's extensibility and tool integration features allow agents to connect with external tools, functions, APIs, or custom code, expanding their capabilities and allowing them to interact with a wider range of systems and data



interact with a wider range of systems and data sources.

The example demonstrates an agentic workflow where `user_proxy` interacts with `executor`, a PythonCodeExecutorAgent. This agent can execute Python code, allowing `user_proxy` to request the execution of a script that prints the Fibonacci sequence up to 100. This showcases how agents with specific

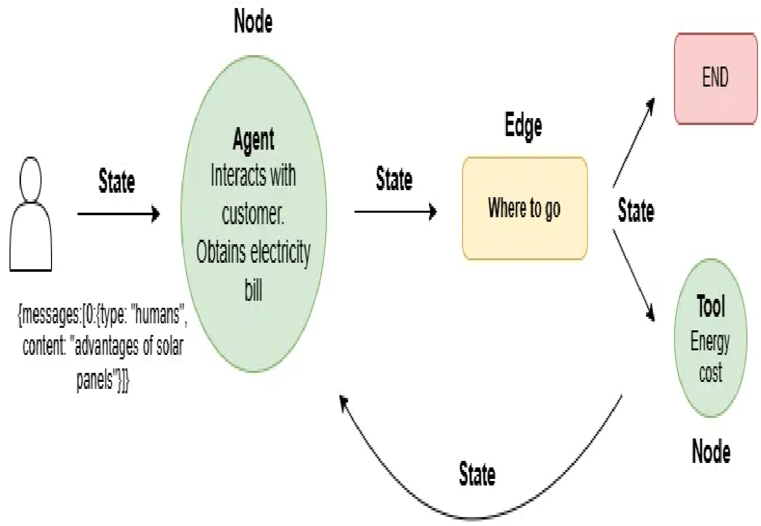
to 100. This showcases how agents with specific capabilities can be combined to achieve complex tasks.

CrewAI allows you to define agents with specific roles and goals, like a "Research Specialist" to find information and a "Scientific Writer" to summarize it. These agents can be orchestrated in a sequence, similar to a production line, to accomplish complex tasks. For example, you could have a

complex tasks. For example, you could have a researcher agent gather information on quantum computing and a summarizer agent then create a blog post explaining it to beginners. This "agentic workflow" is beneficial for use cases like content creation, document processing, and automated research

document processing, and automated research workflows.

Agentic workflow refers to a system that leverages multiple modules, not necessarily all LLMs, to produce a better response to a given query. One method within agentic workflow is compounded LLM, where an initial draft generated by an LLM is critiqued by another LLM, and then refined by a third LLM



by another LLM, and then refined by a third LLM incorporating both the draft and critique. This process can involve iterative loops, with human feedback acting as a critic at certain stages. The ultimate goal of agentic workflow is to enhance the quality and accuracy of the final response through

and accuracy of the final response through collaborative refinement.