How developers really use AI

A survey on software teams and LLMs

```
lass Program

    static void Main(string[] args)
    {
        bool endApp = false;
        // Display title as the C# console calculator app.
        Console.WriteLine("Console Calculator in C#\r");
        Console.WriteLine("------------------------\n");

        while (!endApp)
        {
            // Declare variables and set to empty.
            // Use Nullable types (with ?) to match type of System.Console.ReadLine
            string? numInput1 = "";
            string? numInput2 = "";
            double result = 0;

            // Ask the user to type the first number.
            Console.Write("Type a number, and then press Enter: ");
            numInput1 = Console.ReadLine();

            double cleanNum1 = 0;
            while (!double.TryParse(numInput1, out cleanNum1))
            {
                Console.Write("This is not valid input. Please enter a numeric value: ");
                numInput1 = Console.ReadLine();
            }

            // Ask the user to type the second number.
            Console.Write("Type another number, and then press Enter: ");
            numInput2 = Console.ReadLine();

            double cleanNum2 = 0;
            while (!double.TryParse(numInput2, out cleanNum2))
            {
                Console.Write("This is not valid input. Please enter a numeric value: ");
                numInput2 = Console.ReadLine();
            }

            // Ask the user to choose an operator.
```

Does AI help or hinder software engineering?

Some of the most promising and advanced applications for Generative AI are in software development. LLMs adapt well to coding languages because they're designed to parse and generate structured text, making them naturally compatible with programming languages and tasks.

We surveyed developers on their use of LLMs (namely, ChatGPT) for debugging and general development tasks. We also asked about their experiences of the tasks LLMs can't help with and the functions they'd like to add to the tool to extend it. Here's what they told us, and what we can conclude from the responses.

100%

of respondents reported using ChatGPT in their debugging workflow.

23%

of developers reported challenges with ChatGPT's accuracy and relevance.

35%

noted shortcomings in ChatGPT's technical understanding.

Different respondents named specific tasks, like writing unit tests, as being the most and the least successful use of ChatGPT, reflecting the wide-ranging quality of output based on prompting choices.

The first question for our respondents: How do you use ChatGPT in debugging?

We've separated responses by theme and shown how often they occurred across all responses, expressed as a percentage. Note that some responses contained multiple themes.

Identifying and fixing logical errors (5.56%): Devs seeking workarounds for logical flaws in their code said that they describe the error and outcomes to ChatGPT, discussing the problem encountered and asking for solutions.

> ***"I like to paste in problematic code. It is good at finding logical flaws in code, but not very good at identifying/fixing problems based purely on stack traces (unless provided source code for context)."***

Correcting syntactical errors (5.56%): This includes pasting snippets into ChatGPT and asking for help with fixing typos, syntax errors, or other straightforward coding mistakes.

Understanding and interpreting errors (44.44%): This was the most popular use of ChatGPT for debugging. Almost half of our responses mentioned pasting error messages or exceptions into ChatGPT to gain insight into issues they don't understand, whether this involved trying to decipher an exception, interpreting stack traces, or explaining error messages from third-party applications.

> ***"If I don't recognize the error or don't know what could be the cause I use ChatGPT to get possible causes."***

General debugging assistance (27.78%): These responses refer to broader debugging tasks, such as asking general questions about errors, using ChatGPT as a Google replacement, and getting suggestions on how to use tools more effectively. This category also covers using ChatGPT to understand what scripts are doing or to troubleshoot script failures.

Enhancing and refining code (11.11%): In these responses, users said they asked ChatGPT to identify bugs in larger blocks of code (like methods or classes) and suggest improvements, including refactoring code to handle new scenarios or making it more robust.

Privacy and security precautions (5.56%): This is more of a side effect than a purpose. Several responses mentioned taking precautions to ensure no sensitive or identifiable information is shared with ChatGPT, sanitizing data before input.

Our first notable finding was that all of our respondents reported using ChatGPT for debugging. That could be partly caused by the way we framed the question (basically assuming that they'd at least tried to debug with the LLM), but it's still significant that none of the respondents answered that they didn't use ChatGPT for debugging at all.

Our respondents' biggest use of ChatGPT in the debugging process was for clarifying error messages, exceptions, and issues with third-party applications. They also reported often using the tool for "general debugging assistance", covering multiple use cases, reflecting ChatGPT's versatility for debugging and understanding problems. At the other end of the spectrum, activities like refining code or addressing syntactical errors show respondents using the LLM for more targeted problem-solving in their debugging process.

Next, we asked what other development tasks or processes they found ChatGPT useful for.

Our group of devs reported a range of other uses for the LLM. One disclaimer here is that although our question was about ChatGPT, we also received responses about other AI tools that our respondents reported as useful in their development work.

Documentation and best practices (3.85%): This includes seeking information within the documentation for a given framework or language, understanding best practices, and bouncing proposed architectures or standards off ChatGPT as you would with a peer.

> ***"Highest return for me is not having to look through vast documentation for specific frameworks and programming languages. I know the concepts but each one does things slightly differently."***

Writing and simplifying code (26.92%): Covers writing unit tests, simplifying SQL queries, simplifying or refactoring code, writing quick scripts, and using tools like GitHub Copilot for boilerplate code and tests and writing small code snippets.

Learning and understanding new technologies (15.38%): This includes getting broader overviews or explanations of new technologies or concepts, understanding how to use packages, and generally breaking down how to solve technical problems.

> ***"If there is something I want to do with a package I'll ask how, eg I wanted to register a service with AutoFac in a specific way but didn't know if it was possible, so rather than checking docs I just asked ChatGPT."***

Content creation and communication (15.38%): This entails writing blog posts and other content, message refinement, and customer support. Multiple responses did note challenges with obtaining their preferred tone when using ChatGPT this way (more on that later).

Project planning and development assistance (26.92%): This includes exploring programming ideas, database modeling, script creation, technology stack questions/comparisons, data analysis, API creation, code refactoring, and reviewing PRs.

Efficiency and convenience (11.54%): Several responses mentioned using ChatGPT as a Google replacement, avoiding the need to navigate through resources to find answers, and performing complex tasks quickly, such as managing SSH keys or fact-checking AWS service features.

The most important uses our devs reported were writing and simplifying code and getting help with project planning and development, both mentioned in about a third of the responses. This suggests that ChatGPT is valued for its ability to assist with coding tasks and development planning, as well as for improving efficiency and facilitating learning about new technologies and best practices.

Then, we got controversial. We asked about the tasks developers had unsuccessfully attempted to streamline with ChatGPT.

Here are the jobs or processes that our devs said the LLM just couldn't help with.

Understanding technologies (35.29%): Developers found limitations in ChatGPT's understanding of different technologies, version-specific features, and the distinctions between similar frameworks.

Content creation challenges (11.76%): This includes instances where users tried using ChatGPT to create documentation, blog posts, or writing tasks, but discarded the output when it sounded obviously AI-generated or "salesy".

> ***"Getting it to write a documentation page or blog paragraph never works for me, because it sounds so obviously AI-written and I'm quite picky about my words."***

Testing and development process issues (17.65%): Reflects on challenges related to writing unit tests, evolving code bases, checking whether code is optimized for performance, and the iterative nature of getting to the correct code snippet or script. (The mention of unit tests is pretty interesting here, as another respondent specifically mentioned how well ChatGPT writes unit tests when provided with enough context.)

Accuracy and relevance of information (23.53%): Not a specific task, but these responses noted cases of information provided by ChatGPT being outdated, incorrect, or not detailed enough for meticulous tasks like error resolution, fact-checking, or understanding software internals.

Efficiency in task automation (11.76%): Instances where automating tasks with ChatGPT turned out to be less efficient than expected, either due to the quality of the output or the lengthy iterative process required to refine the output.

> ***"To perform a medium-level task, I used to get ChatGPT to write a Python script to automate it for me. However, I had mixed results with this. Sometimes, the time it takes to get the right output is longer than it would have taken to just write it out myself... This somewhat carries over to writing normal code as well."***

The most frequently reported limitations were poor grasp of niche technical concepts, particularly around specific features, versions, and frameworks, making up over a third of the responses. Accuracy and relevance of information were another big concern, with repeated challenges in obtaining up-to-date and accurate data. Testing and development process issues, as well as efficiency in task automation, also posed challenges, and the overly verbose and banal tone of written content.

Finally, we got their wish list. We asked what capabilities the developers would add to the LLM if they could. Here's what they told us:

Integration with codebases and development tools (44%): Respondents really wanted the LLM to understand more about the context of their queries to give them better responses. They expressed a desire for ChatGPT to automatically know what they're working on, including engineering standards, conventions, and codebase specifics, or directly integrate with the codebase for more precise, context-aware queries.

> ***"I really want to point it at my code directory, ask for system-wide changes, and have them reflected in my browser."***

They also referenced the desire to host a self-contained model that includes the entire codebase and infrastructure for querying system specifics. Similarly, they wanted ChatGPT to have access to repository-level knowledge, plus documentation and communication channels like Slack. Multiple responses also wanted enhancements in handling source code for debugging without having to manually specify the potentially responsible code.

Direct interaction and automation within development environments (22%): The idea of increased integration also extended to the ability to execute system-wide changes directly from a prompt without manual intervention.

Our respondents also wished ChatGPT could handle larger pieces of code more effectively, allowing them to paste in larger snippets and automatically adjust specific code elements.

Expansion of creative and research capabilities (22%): Respondents wanted to add improved research capabilities, essentially having the LLM perform searches of online sources to retrieve current information or data outside of base knowledge.

On a more creative note, one respondent wanted the LLM to create videos, more customizable images according to specific requirements (like dimensions and color palettes), animate images, modify image sections, and generate music.

> ***"It would be nice if it could do the job of googling for me, in cases where I need to find current information or things it doesn't know about."***

Physical interaction (6%): Connecting ChatGPT to a mechanical arm for performing manual tasks to bridge digital and physical workspaces.

Enhanced text handling (6%): Improvements in text management, such as handling larger text inputs or code snippets and possibly altering text based on user-defined rules.

This response had just one clear common theme, with nearly half the answers mentioning the desire for greater integration and context-awareness from the AI and the other responses scattered pretty broadly.

The bottom line

Developers are using ChatGPT for debugging and deciphering errors, writing and refining code, and project planning. They report that the LLM is best for interpreting complex error messages, pinpointing anomalies, and as a research aid, serving as a versatile software development assistant.

Key limitations include difficulties with understanding specific technologies, technical content creation, and efficiently automating tasks. These challenges highlight areas where ChatGPT's capabilities can be improved, particularly in grasping niche technical concepts and generating more tailored outputs.

Overwhelmingly, developers were keen to see deeper integration with development tools and environments, enabling ChatGPT to offer more context-aware advice and automate tasks more effectively. There's also interest in expanding ChatGPT's abilities to support creative tasks and perform more reliable and current research.

So, while ChatGPT seems to be widely used and valued for its ability to clarify technical issues and support various stages of software development, there are gaps. This means there are opportunities to build connections between LLMs and the development workflow and expand their utility across both technical and creative domains.

See all the responses as a spreadsheet here.