Using AI tools for any of the following	CONCLUSION: Using AI tools can improve overall productivity of an engineer (in terms of features implemented per unit time) by maximum of up to 35%. QUESTION: Did you experience:
Generate small snippets of code, Generate tests, Ask AI tools to modify existing code	higher productivity improvements
Generate code for full applications (or large parts of application), Generate small snippets of code, Generate tests, Generate CI/CD scripts, Ask AI tools to modify existing code, Ask AI tools to fix bugs	the same productivity improvements
Generate small snippets of code, Ask AI tools to modify existing code, Ask AI tools to fix bugs, Learn	higher productivity improvements
Generate small snippets of code, Generate tests, Ask AI tools to modify existing code, Ask AI tools to fix bugs	higher productivity improvements
Generate small snippets of code, Generate tests, Ask AI tools to modify existing code, Ask AI tools to fix bugs	the same productivity improvements
Generate small snippets of code, Generate CI/CD scripts, Ask AI tools to modify existing code, Ask AI tools to fix bugs	higher productivity improvements
Generate code for full applications (or large parts of application), Generate small snippets of code, Generate tests, Generate CI/CD scripts	no productivity improvements
Generate small snippets of code, Brainstorming some ideas and ways before writing the code; checking if there are ways that some framework/tool has already a way to deal with certain problems instead of going deep through their documentation first.	lower productivity improvements
Generate code for full applications (or large parts of application), Generate small snippets of code, Ask AI tools to modify existing code	higher productivity improvements
Generate small snippets of code, Ask AI tools to modify existing code, Ask AI tools to fix bugs	the same productivity improvements
Generate small snippets of code	lower productivity improvements
Generate small snippets of code, Generate tests, Ask AI tools to modify existing code	lower productivity improvements
Ask AI tools to modify existing code, support with the development of text	higher productivity improvements
Generate small snippets of code	higher productivity improvements

Generate code for full applications (or large parts of application), Generate small snippets of code, Generate tests	lower productivity improvements
Generate small snippets of code, Ask AI tools to modify existing code, Ask AI tools to fix bugs, Refactoring (give example of existing code and give it example of how I want the updated code to look like, then give it existing code in snippets. Useful when refactoring code that looks the similar multiple times	higher productivity improvements
Generate small snippets of code	the same productivity improvements
Generate code for full applications (or large parts of application), Generate small snippets of code, Generate tests, Ask AI tools to modify existing code, Ask AI tools to fix bugs	higher productivity improvements
Generate small snippets of code, Check solution for error, write commets to methods	the same productivity improvements
Generate small snippets of code	the same productivity improvements
Generate small snippets of code, Ask AI tools to modify existing code, Ask AI tools to fix bugs	the same productivity improvements
Generate small snippets of code, Ask AI tools to modify existing code, Ask AI tools to fix bugs	the same productivity improvements
Generate small snippets of code, Ask AI tools to modify existing code	higher productivity improvements
Generate small snippets of code, Ask AI tools to modify existing code, Ask AI tools to fix bugs	the same productivity improvements
Generate small snippets of code, Generate tests, Ask AI tools to modify existing code	higher productivity improvements
Generate small snippets of code, Generate tests, Ask AI tools to modify existing code, Ask for different approaches for solving problems	the same productivity improvements
Generate small snippets of code, Generate tests, Ask AI tools to modify existing code, Ask AI tools to fix bugs, explaining legacy code	higher productivity improvements
Generate small snippets of code, Generate tests, Ask AI tools to fix bugs	the same productivity improvements

Generate code for full applications (or large parts of application), Generate small snippets of code	higher productivity improvements
Ask AI tools to modify existing code	higher productivity improvements
Generate small snippets of code, Ask AI tools to modify existing code, Ask AI tools to fix bugs	higher productivity improvements
Generate small snippets of code, Ask AI tools to modify existing code, Generate code comments (e.g. explain the code)	lower productivity improvements
Generate small snippets of code, Generate tests, Ask AI tools to fix bugs	higher productivity improvements
Generate small snippets of code, Generate tests, Generate CI/CD scripts	lower productivity improvements
Generate code for full applications (or large parts of application), Generate small snippets of code, Generate tests, Generate CI/CD scripts, Ask AI tools to modify existing code, Ask AI tools to fix bugs, Ask AI to build visual components / UI (html / css)	much higher productivity improvements
Generate small snippets of code, Generate tests, Ask AI tools to modify existing code, Ask AI tools to fix bugs, Ask AI tools to explain how to use libraries	lower productivity improvements
Generate code for full applications (or large parts of application), Generate small snippets of code, Generate tests, Generate CI/CD scripts, Ask AI tools to modify existing code, Ask AI tools to fix bugs, Documentation	higher productivity improvements
Generate small snippets of code	the same productivity improvements
Generate small snippets of code, Ask AI tools to modify existing code, Ask AI tools to fix bugs	the same productivity improvements
Generate small snippets of code	the same productivity improvements

CONCLUSION: The most beneficial moment to apply AI tools is in the early code implementation stages - to generate a skeleton/basic features for early requirements describing fundamental features of an application. QUESTION: Do you agree:	CONCLUSION: Long processing time of AI tools (time that an AI tool needs to process request and provide an answer) has a significantly negative effect on productivity. QUESTION: Do you agree:
agree	strongly disagree
neutral	strongly disagree
strongly agree	disagree
agree	agree
agree	neutral
neutral	disagree
neutral	agree
agree	neutral
agree	disagree
neutral	neutral
agree	agree
strongly agree	strongly disagree
also for testing purposes	neutral
agree	disagree

strongly agree	disagree
agree	strongly disagree
strongly agree	agree
agree	agree
neutral	disagree
agree	disagree
agree	agree
strongly agree	disagree
disagree	neutral
neutral	agree
agree	disagree
disagree	agree
agree	disagree
disagree	agree

strongly agree	disagree
neutral	neutral
In my experience it can be used throughout the software development lifecycle. I feel it is more useful as the complexity of the code gets higher.	disagree
agree	disagree
disagree	strongly disagree
agree	strongly agree
disagree	disagree
strongly disagree	neutral
agree	disagree
disagree	disagree
agree	strongly disagree
agree	neutral

CONCLUSION: As code size and code complexity increase, productivity gains decrease because AI tools take a longer time to process and return the larger code segments (processing time of AI tools, e.g., waiting for 10 minutes to receive prompt execution)	CONCLUSION: As code size and code complexity increase, productivity gains decrease because AI tools are more likely to break existing code (e.g., need to repeat the prompt 2 to 3 times on average). QUESTION: Do you agree:
neutral	disagree
neutral	agree
neutral	neutral
strongly agree	strongly agree
neutral	neutral
agree	agree
agree	strongly agree
neutral	neutral
agree	agree
agree	agree
agree	strongly agree
Never asked about to a complex code to an AI, ChatGPT free version is not smart enough	strongly agree
neutral	agree
neutral	neutral

strongly agree	agree
disagree	agree
agree	agree
neutral	agree
neutral	agree
neutral	agree
strongly agree	strongly agree
disagree	strongly agree
agree	strongly agree
neutral	agree
neutral	neutral
agree	strongly agree
agree	strongly agree
strongly agree	agree

neutral	neutral
agree	neutral
disagree	disagree
neutral	agree
disagree	neutral
agree	agree
strongly disagree	neutral
disagree	agree
disagree	disagree
agree	neutral
neutral	strongly agree
neutral	agree

CONCLUSION: Asking AI to perform code changes on existing code can be significantly more time consuming (due to communication and the need to create several prompts before reaching the desired effect) than if a human does the same work.	CONCLUSION: Asking AI to perform bug fixes on existing code can be significantly more time consuming (due to communication and the need to create several prompts before reaching the desired effect) than if a human does the same work.
disagree	disagree
neutral	disagree
disagree	Can be good if problem is well explained
agree	agree
agree	disagree
neutral	neutral
strongly agree	strongly agree
agree	agree
neutral	agree
agree	agree
strongly agree	agree
neutral	agree
disagree	agree
agree	neutral

strongly agree	strongly agree
agree	neutral
agree	neutral
I can not express my intention clear enough. AI does not understand my end goal.	If its a small memory leak, than it is easier for AI to find. Similarly as previous, AI does not understand the context
agree	agree
strongly agree	neutral
agree	agree
strongly agree	strongly agree
disagree	disagree
agree	agree
disagree	disagree
agree	strongly agree
strongly agree	agree
strongly agree	strongly agree

agree	agree
agree	agree
disagree	neutral
agree	agree
neutral	agree
strongly agree	disagree
largely depends on the familiarity of dev with design	disagree
agree	agree
neutral	neutral
agree	agree
agree	disagree
agree	neutral

CONCLUSION: The best way to use AI tools for improving productivity is to break a problem into small chunks and ask AI to implement it. Trying to solve bigger problems at once leads to bugs or incorrect solutions. QUESTION: Do you agree:	CONCLUSION: The integration of AI generated code with the rest of the code base takes significant effort (e.g., need to change variable names, function signatures, module interfaces). QUESTION: Do you agree:
agree	neutral
agree	disagree
agree	disagree
agree	neutral
agree	neutral
neutral	agree
strongly agree	agree
strongly agree	neutral
strongly agree	disagree
agree	neutral
strongly agree	disagree
agree	disagree
agree	neutral
agree	agree

strongly agree	disagree
strongly agree	neutral
agree	neutral
agree	neutral
strongly agree	neutral
agree	agree
strongly agree	agree
agree	agree
agree	strongly agree
strongly agree	agree
neutral	disagree
strongly agree	neutral
strongly agree	disagree
strongly agree	disagree

agree	neutral
agree	agree
strongly agree	disagree
strongly agree	disagree
agree	agree
strongly agree	neutral
agree	disagree
strongly agree	disagree
neutral	neutral
strongly agree	agree
strongly agree	neutral
agree	neutral

Do you have any other comment on how AI tools improves productivity?
I mostly use ClaudeAI, as its responses for code and development-related questions are generally better than ChatGPT's. I don’t use any copilot tools.
Brainstorming ideas and asking the AI first for some ways to implement the code in a certain framework/library instead of going through their documentation first can save time.
Explaining the problem to be solved to the AI forces you to understand and collect the requirements. It somehow paves the way for good requirements engineering
Mostly reliable for boilerplate code and simpler templates.

Relieves the developer from lengthy boilerplate code generation.

Repetitive work when it comes to refactoring similar code.

It is very fast for writing obvious and repetitive code. However it always needs careful revision as it may still make surprising mistakes.

1. Helps jump into a topic more easily. 2. Helps with regex, string functions, etc.

creating documentation, helps with naming

Our AI team uses LLMs to quickly sum up and recommend the latest cutting-edge research that’s relevant to our work, which really speeds up getting new tech into our apps. This has actually been more useful to us than help with coding.

One advantage I see in using AI tools is the ability to quickly access information from design documents and other company-specific resources. Previously, finding and compiling the needed information was time-consuming. With AI tools trained on the company’s existing documents, it’s now much faster and easier to retrieve relevant information.

It can help with documentation, explaining why certain output results from code at hand

The main productivity improvement I noticed is the fact of being able to ask questions like "how do I do this?" rather than asking the tool to output code.

Depending on the tool and the wording you use, results in productivity may be very different.

Do you have any other comment on challenges with AI tools when it comes to the productivity?
Blindly trusting the AI code can have a negative impact on the productivity. Decent understanding of a programming language before using AI for coding would be recommended as it is way easier to spot some things that wont work or are off that the AI provides (AI tools tend to even make things up just to return some answer)
Anything significantly complex usually takes multiple attempts and can even negatively impact performance.

After couple of prompts it forgets some of the instructions given earlier.

AI should be used like honey. A spoon per day.

It decreases productivity for beginners and juniors because they are most likely to use AI generated code that they don't understand. It is almost not productive at all. If they would like to use it, if should be just for learning and research, for some questions like they would ask a mentor.

does not understand context of big projects

The reasoning and comprehension abilities of current LLMs are still limited, setting a ceiling on what they can achieve. For example, they can be very helpful for junior engineers with tasks like fixing simple bugs or writing a timezone conversion function correctly on the first try. But when it comes to system architecture design, high-level performance optimization, or complex bug resolution, they provide almost no assistance. Imagine asking Copilot to help Linus optimize Linux’s file system—it’s just not feasible. AI is improving, but this is still the reality.

AI tools often produce wrong results and have to be told their output is incorrect several times before being really helpful

We are still discovering how to integrate tools in our workflows. It's still early to know the final form of the most productive pattern (on average).

The generated skeleton codes are helpful but they need to be reviewed carefully to avoid bugs. Sometimes, reviewing the generated code takes longer than writing the code piece yourself.

Compliance with company standards on (use of) AI

CONCLUSION: When generating full applications or large portions of code involving multiple functions/classes at once, partition and organisation of AI-generated code can be of a significantly lesser quality to that of human-written code.	CONCLUSION: Using smaller snippets of AI-generated code/requesting smaller changes does NOT lead to a significant erosion of software architecture (the code is of high cohesion, low coupling, human-alike separation of logical parts).
disagree	agree
neutral	strongly agree
neutral	neutral
not applicable in my case	agree
neutral	agree
neutral	strongly agree
strongly agree	neutral
neutral	agree
neutral	strongly agree
agree	agree
agree	agree
disagree	neutral
disagree	strongly agree
neutral	agree

strongly agree	agree
not applicable in my case	neutral
disagree	agree
agree	agree
agree	agree
agree	agree
strongly agree	disagree
agree	agree
agree	neutral
not applicable in my case	neutral
not applicable in my case	agree
not applicable in my case	agree
agree	strongly agree
strongly agree	agree

neutral	strongly agree
agree	neutral
disagree	agree
not applicable in my case	agree
agree	agree
neutral	agree
agree	strongly agree
not applicable in my case	not applicable in my case
strongly disagree	agree
agree	neutral
agree	agree
not applicable in my case	neutral

CONCLUSION: Code generated by AI is of comparable performance quality (optimized for performance, no significant unnecessary overhead) to that of human-written code. QUESTION: Do you agree:	CONCLUSION: Code snippets (smaller code chunks) generated by AI are of comparable maintenance quality (high cohesion, low coupling) to that of human-written code. QUESTION: Do you agree:
agree	neutral
agree	agree
neutral	neutral
neutral	neutral
agree	agree
agree	agree
neutral	neutral
neutral	neutral
strongly agree	disagree
neutral	agree
Not necessarily. You have to specify that you require performance optimizations, and even still have to put in some human touch to it.	neutral
agree	agree
agree	disagree
agree	agree

neutral	strongly disagree
neutral	agree
strongly agree	strongly agree
agree	agree
neutral	neutral
agree	agree
agree	neutral
neutral	agree
disagree	disagree
agree	agree
agree	agree
disagree	agree
disagree	agree
agree	agree

neutral	agree
neutral	agree
disagree	strongly agree
agree	agree
agree	neutral
agree	agree
strongly agree	strongly agree
disagree	agree
agree	agree
disagree	neutral
agree	agree
neutral	neutral

CONCLUSION: AI-generated code is almost always syntactically correct. QUESTION: Do you agree:	CONCLUSION: AI-generated code rarely fulfills desired functionality in the first shot (true for small or big code chunks). As the problem size increases, so does the need for fixing the generated code (either through re-generation or a human intervention)
agree	neutral
neutral	strongly agree
strongly agree	disagree
neutral	agree
neutral	agree
agree	agree
disagree	strongly agree
agree	neutral
agree	agree
agree	agree
disagree	strongly agree
agree	neutral
disagree	agree
agree	agree

disagree	strongly disagree
disagree	agree
strongly agree	agree
agree	agree
neutral	neutral
agree	neutral
agree	strongly agree
agree	strongly agree
disagree	agree
neutral	agree
disagree	neutral
strongly agree	strongly agree
disagree	strongly agree
strongly agree	disagree

agree	agree
neutral	agree
agree	neutral
agree	disagree
disagree	agree
disagree	neutral
strongly agree	strongly disagree
agree	strongly agree
agree	disagree
disagree	strongly agree
disagree	strongly agree
disagree	strongly agree

CONCLUSION: Code generated by AI is of comparable logical syntax style (modifiability, consistency and understandability of identifier names, adequate length of code lines) to that of human-written code and therefore easy to understand.	CONCLUSION: Tests generated by AI tend to use more complex testing techniques (e.g., mock objects, pixel colour testing). QUESTION: Do you agree:
agree	disagree
agree	strongly agree
disagree	disagree
agree	neutral
neutral	neutral
agree	neutral
disagree	disagree
agree	neutral
disagree	I've never used AI for testing, thus I cannot answer it
neutral	neutral
neutral	strongly disagree
neutral	agree
agree	agree
agree	neutral

agree	agree
agree	neutral
It is probably better than the average programmer	I have not generated tests with AI
Still did not use AI for tests	Still did not use AI for tests
disagree	disagree
disagree	agree
agree	can't answer - haven't focused on tests
neutral	neutral
strongly agree	neutral
agree	neutral
agree	agree
agree	agree
agree	agree
strongly agree	neutral

disagree	neutral
agree	neutral
agree	neutral
agree	neutral
disagree	disagree
neutral	disagree
strongly agree	neutral
neutral	strongly disagree
agree	agree
neutral	disagree
neutral	neutral
agree	no experience

CONCLUSION: Around 50% of generated tests initially fail, due to the low quality of tests. QUESTION: Do you agree:	Do you have any other comment on how code generated by AI tools affects quality of software design architecture?
disagree	
disagree	
neutral	
agree	
neutral	
neutral	
neutral	
neutral	
I've never used AI for testing, thus I cannot answer it	
neutral	
agree	AI tends to generate tests to make code succeed, I.E. does not generate tests based on requirements, so it's basically a waste of time for anything significant.
disagree	
disagree	
neutral	

neutral	
neutral	
I have not generated tests with AI	
neutral	
neutral	
neutral	
can't answer - haven't focused on tests	
neutral	
neutral	
neutral	
neutral	
agree	
agree	
agree	the prompts really matter

neutral	
neutral	
neutral	
neutral	
neutral	
agree	
neutral	
agree	
neutral	Again, results depend highly on the ability of the user to prompt the tools correctly.
neutral	
neutral	
no experience	

AI tools that you use in your daily work	What is your current role?	How many years of experience do you have in software engineering?
ChatGPT chat bot	Software Architect	More than 6 years
ClaudeAI	Software Developer	More than 6 years
CoPilot	Team lead	More than 6 years
CoPilot, ChatGPT chat bot	Software Developer	More than 6 years
CoPilot, ChatGPT chat bot	Software Architect	More than 6 years
Claude	Software Developer	Less than 3 years
ChatGPT chat bot	Software Developer	Between 3 and 6 years
ChatGPT chat bot	Software Developer	Between 3 and 6 years
ChatGPT chat bot	Software Developer	More than 6 years
ChatGPT chat bot, Cursor IDE	Software Developer	Less than 3 years
ChatGPT chat bot	Software Developer	Between 3 and 6 years
CoPilot, ChatGPT chat bot	Software Developer	Between 3 and 6 years
ChatGPT chat bot, grammarly GO	Software Developer	Between 3 and 6 years
CoPilot	Software Developer	More than 6 years

ChatGPT chat bot	Software Developer	More than 6 years
CoPilot, ChatGPT chat bot	Software Developer	Less than 3 years
CoPilot, Claude Dev	Project Manager	More than 6 years
ChatGPT chat bot, claude.ai	Software Architect	Less than 3 years
ChatGPT chat bot	Software Developer	More than 6 years
ChatGPT chat bot	Software Developer	More than 6 years
ChatGPT chat bot	Software Developer	Between 3 and 6 years
CoPilot, ChatGPT chat bot	Software Developer	Between 3 and 6 years
CoPilot, ChatGPT chat bot, GPT Pilot, grammarly GO	Team lead	More than 6 years
ChatGPT chat bot	Team lead	Between 3 and 6 years
CoPilot, ChatGPT chat bot, grammarly GO	Researcher	Between 3 and 6 years
ChatGPT chat bot, GPT Pilot	Software Developer	Between 3 and 6 years
CoPilot, Amazon Q	Team lead	More than 6 years
CoPilot, ChatGPT chat bot	Team lead	More than 6 years

CoPilot, ChatGPT chat bot	Team lead	More than 6 years
CoPilot, ChatGPT chat bot	Software Developer	Between 3 and 6 years
CoPilot, ChatGPT chat bot, GPT Pilot	Software Developer	More than 6 years
CoPilot, ChatGPT chat bot, GitHub Copilot	Software Developer	More than 6 years
CoPilot, ChatGPT chat bot	Software Developer	More than 6 years
CoPilot, ChatGPT chat bot	Team lead	More than 6 years
ChatGPT chat bot, Cursor IDE, claude.ai / v0.dev	Software Architect	More than 6 years
CoPilot, ChatGPT chat bot	Software Developer	More than 6 years
CoPilot, ChatGPT chat bot, Gemini (formerly Bard), Cursor IDE, open-source projects and personal tooling	Engineering Manager, Software Architect, Sales Solutions Lead, Programs Lead	More than 6 years
ChatGPT chat bot	Software Developer	Less than 3 years
CoPilot, CodeWhisperer	Team lead	More than 6 years
Gemini (formerly Bard)	Team lead	More than 6 years

How many employees does the institution where you work have?	Country where your office is based
50-499	Bosnia and Herzegovina
50-499	Germany
500 - 4999	Germany
500 - 4999	Germany
500 - 4999	Germany
1-49	Bosnia and Herzegovina
1-49	Bosnia and Herzegovina
1-49	Bosnia and Herzegovina
>5000	Germany
1-49	Bosnia and Herzegovina
50-499	Bosnia and Herzegovina
1-49	Germany
1-49	Germany
>5000	Austria

500 - 4999	Germany
1-49	Bosnia and Herzegovina
1-49	Australia
>5000	Germany
1-49	Germany
1-49	Bosnia and Herzegovina
50-499	Finland
500 - 4999	Bosnia and Herzegovina
50-499	Norway
1-49	Norway
500 - 4999	Norway
1-49	Serbia
>5000	Austria
50-499	China

1-49	Germany
1-49	Bosnia and Herzegovina
>5000	The Netherlands
>5000	Serbia
50-499	Germany
>5000	Austria
50-499	Germany
500 - 4999	Spain
50-499	Dominican Republic
500 - 4999	Sweden
>5000	UK
>5000	Netherlands