The columns represent the following:
Participant ID: Identifier for the participant providing the data.
Raw data: Participant responses.
Code: Initial labels assigned to segments of raw data
Concept: Grouping related codes.
Sub-category: Grouping related concepts.
Category: Grouping related sub-categories.

Participant ID	**Raw Data**	**Code**	**Concept**	**Sub-category**	**Category**
P3	P3: Coding related… So if we're talking about coding related tasks. I think it's mainly an idea generator for me when given a task or feature to develop. Usually you want to brainstorm on what the potential approach or solutions for that specific feature can be. So you would use ChatGPT, to generate those ideas for you, and then see if it aligns with what you want. It sort of just speeds up the process rather than you having to think through it yourself. And then, in terms of debugging, **I think ChatGPT can be useful for debugging if you feed it, and the exact error that you're facing. So for sometimes, when you're developing something, and the error message pops up rather than searching Google for that specific error message and going through stack Overflow forums to try and figure out what the issue is, I think ChatGPT is pretty good at if you just gave it that chunk of error message.** It'll somehow know what is probably wrong. And then usually, I would follow the debugging steps that they laid out for me. And then sometimes that works. Sometimes it doesn't, as usual. And also for the test driven development stuff, I think, like I mentioned it, If you feed in a test scenario that you want, it can give you sort of a lot of options that you can try out.	LLMs Speeding Up Debugging Process	LLMs Boosting Code Development	Impact On Productivity	Impact of using LLMs
P1	**P1: I used it a lot for Java.** In Java, I used it a lot **to make the process faster and other languages that I don't have much mastery of, like Python**. I don't know Python very well. So it was him who helped me the most to be able to program some specific things that you have to do in Python, for example. Java and Python, I think those two are the main ones.	LLMs Supporting Experienced Developers Coding In Non-Familiar Programming Languages	LLMs Boosting Code Development		
			LLMs Empowering Developers		
P8	Researcher: And when you think about the influence that LLMs has on developers career, what… I mean is like going through educational and training and then jumping into seniority levels from juniors, mid level, seniors and so on… How do you view LLM's influence the developers career? And how do you feel that the job market is also influenced in a way of become more competitive? P8: So, I [can] think one for sure it's: **if you do not use AI** - I think this is like a [difficult decision] to some developers who are maybe because of political or maybe personal value, they don't like to use AI for that reason **- they might be left out [by those who adopt AI tools]**. And because the AI nowadays is just another tool, that may be my failure, that I said many times… The value, **the perspective that I hold here is: AI is a tool, and if you don't use the right tool for the right context or right situation, then you might be left out [by those who adopt AI tools], because AI is definitely [something that] can speed things up**. And having said that, and from what I have seen, so far from the education and training perspective, a lot of people start with AI, and then they fully trust it. And then, when they got criticism or feedback from more experienced [developers], they start losing that [over] trust, and they started to be more careful in using it and [then] they start to learn it. I think that's also my experience so far in using AI which is like [starting] fully trusting, and then starting to reduce that trust until I find the right balance, which is, for now I use it to help me understand the codes. Yeah, [my use of AI tools is focused on making them] help me to understand the codes mostly. And also, although I asked it to generate it, make sure that when it is generated, I still understand what's going on. And so, in that sense, the seniority levels here maybe they can help on [dealing with it]. Because the more senior you are, most [of your work] likely [is] just gonna be [doing] the code reviewers. And when you are doing that code reviews, you can maybe help AI to come up with some kind of recommendations on what areas that [AI] need to be looked at. And maybe there are some bad practices here and there [in the codes generated using AIs], and then maybe give suggestions [on how to fix/adjust]. And the seniors can maybe use that. That's why, in terms of job market competitiveness, if you are [one] those who do not want to use AI, maybe that's fine, because that's your value. But those who want to use AI as a tool, I think they will be much more competitive or much more competent than others who don't [use AI tools at all]. That's my opinion.	Losing Gains in Speed by Not Adopting LLM tools	Sacrificies by Avoiding LLMs	Impact On Productivity	Impact of using LLMs

The columns represent the following:
Participant ID: Identifier for the participant providing the data.
Raw data: Participant responses.
Code: Initial labels assigned to segments of raw data
Concept: Grouping related codes.
Sub-category: Grouping related concepts.
Category: Grouping related sub-categories.

Participant ID	Raw Data	Code	Concept	Sub-category	Category
P18	Researcher: What people lose by not adopting those tools? **P18: I think we can make things faster by using ChatGPT and AI tools. But without using that we are not that fast, and we will lose our ties, mostly not adapting for those.**				
P8	Researcher: Have you ever come to any limitations while using Llms? Not just challenge, but limitations. P8: Yes, the biggest limitation here is… let's say that Perplexity.AI [‘s limitation] - I haven't really used Copilot, GitHub Copilot, but with Perplexity.AI here - is when we have more files, more systems, more classes - things that are connected to each other - Perplexity.AI lacks understanding of how things are connected. That's the first [limitation]. The second is: **we have to guide [Perplexity.AI] quite a lot in terms of principles [of software engineering] and quality - [things] that we learn as a software engineer. Otherwise, it's just gonna generate rubbish. I always say [that the code generated via AI is a] Frankenstein, because it's always mixing and matching different things together, without fully understanding the full system.** Maybe if it generates from scratch [with no dependency on any code] - I'm not sure - maybe it should be fine - but [using] from the beginning of the project - and then make it allow this to generate it as a full system, fully understood by that AI. But in most of my cases, I always have [this situation where] someone has already built this [software components previously]. And then [when I use] this AI trying to help it, I always just use it as a tool. If I fully trust this to AI, [the code] may not fully work, especially when we talk about networks, like when we have to connect different modules and microservices. I don't think it [LLMs] understands all different dockers and different systems, how they are connected to, etc. Maybe it doesn't understand that well.	Need A lot of Effort guiding LLMs to Maintain Good Code Quality	LLMs Slowing Developers Increasing Effort due to LLMs	Impact on Developers' Effort	
P4	P4: Limitations... S**ometimes I don't feel willing to [write a prompt]. Then there's something I know what to do, I know how to do it, but I'm also too lazy to explain to the prompt what I want**. **Then I think: “hmm... I won't know how to write and it [LLM] will change things I don't want. I'll have to copy and edit, copy and edit, until I get what I want. Then I'll do it the way I want”.** Then I go and do it by myself. But usually it's like this, for example: I'm going to create a flow of something from scratch. Then I [say to the LLM]: “I want a flow that does this and that, but has this condition”. I get lazy when it's a very complex prompt. I'm too lazy [to write the prompt], I prefer to do it myself. Generally, the things I ask of them [LLMs] are either very generic - for example, give me a class that will take all the logs and send them to a business - It's something very generic. Then he [LLM] gets it. It's generic, but it's for a specific thing. It doesn't have many business rules that are in my context. It's more generic. I think it can handle it better. But when it's something more in my context that has a lot between the lines, that I have to be thinking about in order to solve, I'm too lazy to write it down. I'm too lazy to write a long text for an IA. I do it using my head.	Feeling Lazyness to Write Complex Prompts for FamiliarTasks	Influence of Developers' Personality	Attitudes & Perceptions	Mindset

The columns represent the following:
Participant ID: Identifier for the participant providing the data.
Raw data: Participant responses.
Code: Initial labels assigned to segments of raw data
Concept: Grouping related codes.
Sub-category: Grouping related concepts.
Category: Grouping related sub-categories.

Participant ID	Raw Data	Code	Concept	Sub-category	Category
P2	Researcher: Yeah, so could you tell me about your experience? what made you try as well. P2: Yeah. So the way **I started using it [LLM/Perplexity.AI] initially was to look up documentation of things, because that was very convenient. Otherwise I'll have to, you know, find the official docs, and then go and read a bunch of talks and stuff. So I found it much easier to just use a chat, use an Llm to look up for the documentation of particular functions and things like that.** So what I did was, let's say, if I had a particular functionality to do like. If I wanted to look up the function that would give me a random number between, you know, let's say M, and then, so I will just put that as a prompt in the Llm. And I would get a function, and then just get the description of what this function does, and all that. So that was very convenient for me. But at the same time I didn't use it too much like a Github copilot thing like I always used to use this as a reference and not as a coding tool. because I sort of felt like, you know, I'll be learning better when I'm using it as a reference rather than you know, just blindly copying things from it. So that's how I initially used it most of the time. And, to be honest, I started using this on I outside of work rather than at work, because we were told to not use it because of all the licensing issues and all the things that might be there. So I started using it on the side initially. When I was working back then, and also when I'm doing my assignments, I used it as a reference. And also I have used it to generate code in some cases where the logic was pretty tricky, and I just wanted to see how it would handle things. but it has been a hit or miss. Sometimes it produces accurate, really good code, but at the same time, sometimes it'll be actually really bad, and it might look like as if you're getting a good output. But when you try to run it. It's [the output is] actually not that good. So that has been my experience pretty much.	Chat On Perplexity is Easier Than Manual Search on Official Documentation	LLMs Reducing Developers' Effort	Impact on Developers' Effort	Impact of using LLMs
			LLMs Saving Time	Impact On Productivity	
P6	P6: Yes, it is. Pair programming with people is more dynamic. It's easier for you to explain it so that someone else can help you, but... it's difficult to explain [the difference between pair programming with ChatGPT and with people]... I think both have their advantages and disadvantages. ChatGPT is much faster in this sense, it's much more dynamic. Pair programming with it is very good for improving your code, delving into something in the code that you want. But I feel that from my experiences so far, when I do pair programming with a normal person, for example, another person from my team, it tends to be richer, because it usually lasts a long time. So, for example... but it also depends on the pair programming [implemented in the company]... At least in my company, we have a very big culture of pair programming. So, we have some pair programming techniques. So I'm kind of being a bit biased. But **all pair programming here is very fruitful in the sense that everyone takes part. They last for hours and you can delve into various subjects.** I feel that pair programming with ChatGPT is much more practical, dynamic and quick: “I've developed this and that's it”**. With other people I tend to develop knowledge in other areas that I didn't even want to know about during the meeting, I didn't even know I needed it, and during the pair programming I learn about it.**	Unexpected Learning Opportunities In Traditional Pair Programming	AI Pair Programing VS Traditional Pair Programming	Impact on Process	

The columns represent the following:
Participant ID: Identifier for the participant providing the data.
Raw data: Participant responses.
Code: Initial labels assigned to segments of raw data
Concept: Grouping related codes.
Sub-category: Grouping related concepts.
Category: Grouping related sub-categories.

Participant ID	Raw Data	Code	Concept	Sub-category	Category
P19	Researcher: Could you compare, for example, pair programming that you have humans working together, there's a goal, and the AI pair programming? Could you compare both? **P19: I will say that in pair programming with human, both of the parties [developers] learns from each other and criticise each other. And in pair programming with AI, I am the only person who is actually criticising the AI, and AI is just doing the worst. Actually, that is not [pair] programming in any sense. It's just AI doing the work.** Researcher: And why do you feel that is not pair programming? P19: So the thing is that AI is not criticizing me [instead of] it is extremely polite. And I am the one person that is doing the criticizing of other person, and the AI is actually doing the work. And AI is actually getting the upper hand here, because that is faster than me. So that's why one party [AI] is actually doing the work and other parties not doing anything. R**esearcher: And what do you feel are the benefits that you have on pair programming that you lose while you're using when you use LLM tools?** **P19: I think that we actually criticise each other. And we actually have a discussion [where] each parties try to establish their discussion their point. And that is a very healthy discussion. We [developers] learn from those kinds of discussions, and I think that is the good part of programming with humans**. But [while interacting] with AI, there is not such kind of discussion or debate, this kind of thing.		Losing Learning Opportunities	Impact on Learning	
P7	Researcher: And how do you think that LLMs influence your career as a bi analyst in the near future, and also in the long term? **P7: Oh, in the near future, I think I would be using ChatGPT a bit more than now. In the next month we're [planning in] doing a lot of dashboard cleanup, and that process requires a lot of [brainstorming,] generating ideas like: "what kind of graph would look good for this kind of data?" which I can use ChatGPT to come up with new creative ideas.** And in the long term, I'm hoping that I will have to use it less. I'm hoping that I will eventually develop the skill sets that I need for my role, [so] that I don't have to go to ChatGPT [because] I will generate the ideas myself.	Increasing Reliance On LLMs for Brainstorming in the near future	Overreliance on LLMs	Reliance on LLMs	Minset
			Influence of Developers' Personality	Attitudes & Perceptions	
			Losing Learning Opportunities	Impact on Learning	Impact of using LLMs
P1	**Researcher: And when you feel that ChatGPT, Copilot or Gemini gives you the wrong answer, what do you do then? What is your approach to identifying it and trying to resolve it?** **P1: I try to see if the wrong answer was because I didn't give enough information. Then I read my prompt, I'll see if the prompt I sent was a correct prompt. If I really did send something that didn't make sense, or if I didn't specify differently, I go there and specify.** Or if he [LLM] really got it wrong, the LLM got it wrong, I'll go and try to correct it. If not, there's nothing to do. I'll try to solve it manually, talk to other people to try to solve the problem.	Searching for Reason Behind Wrong LLM Ouput	Increasing Effort due to LLMs	Impact on Developers' Effort	Impact of using LLMs
			Gaining Learning Opportunities	Impact on Learning	
			LLMs Slowing Developers	Impact on Productivity	