Palestrante 1: And OK, so now we are recording it. Good. So… So look, what's your current position? I will, I will start with some demographic questions just to understand what to do.

Palestrante 2: Yeah.

Palestrante 1:

Palestrante 2: I am a senior software engineer at <omitido>.

Palestrante 1: OK, at <omitido>? Good. So basically you develop code like… you don't… you are not like…

Palestrante 2: Mainly, mainly I develop code and I also take decisions related to… Uh, what? I mean what we are doing right now and in the past… let's say six months, something like this. It's a bunch of refactoring, a bunch of…

Palestrante 1: OK.

Palestrante 2: And so I take design, solution design decisions also. But yeah, mainly my most of my time is spent on coding.

Palestrante 1: Yes. OK. But you also have some management role, like deciding what to do and prioritizing pieces.

Palestrante 2: Yeah, basically, it is not a startup in the sense that we are part of a huge group, but in fact we are a startup still. So basically we work like… It's not like, you know, very narrow. What I do, I can, you know, participate in management meetings related to some aspects, and maybe I can also spend time fixing a bug which could be done… But it's… Yeah, my role is mainly designing and implementing solutions in our software applications.

Palestrante 1: Great, great. And for how long have you been doing that? How many years of experience?

Palestrante 2: Well, I'm at <omitido> since February, so it's like 6–7 months. But I mean, I've been in the industry since, I think, 2015, something like this. So it's 10 years.

Palestrante 1: Around 10 years of experience, great.

Palestrante 2: Right.

Palestrante 1: That's great. That's great. Good. So now, after the demographics, we will move to the actual evaluation. So basically, the evaluation is split into main sections. First, we are going to evaluate the usefulness of the bot—how useful you perceive it to be in your job. The next section is going to be about ease of use—so, is it easy to understand how to operate the bot, and so on? Then you mentioned you had the chance to interact with the bot a bit. Overall, what did you think about it?

Palestrante 2: Well, it was quite simple. I mean, I didn't have any kind of problem or issue during the installation process. I would say to start to have the bot working is easy.

Palestrante 1: Mm-hmm.

Palestrante 2: My opinion is that probably the main… let's say… I mean, where you have to learn a bit more is how the bot works. So, if you have an issue, if you're creating an issue and you know it's a tech debt issue, there are different options. I might have a manual tag that I put on the issue, and then I might also have the bot, so there is kind of… I would say there is a problem with reconciling the two tags—if you have a tag from the bot and a manual tag which is finally used.

Palestrante 1: Mm-hmm. Yes.

Palestrante 2: And both refer to technical debt. Somehow, you have to reconcile, so at least it should be configurable. You should be able to say something like, "OK, I have this manual tag. This is my manual tag. When I put this tag, the bot should decide for… take that action," or you should be able to customize it. As it is, if I have a manual tag, the bot might fail in the classification and say something like "this is not like that," and you might end up with two tags, one saying "this is that" and the other saying "this is not like that."

Palestrante 1: OK.

Palestrante 2: This kind of situation is something I've thought about. It should be customizable—you should decide, "OK, this is my manual tag; what the bot has to do when I put this tag?"

Palestrante 1: Yes.

Palestrante 2: You go on as you want, and I will double-check later if it's like that or not. Or…

Palestrante 1: I see.

Palestrante 2: Conversely, the bot could be right and you should change the manual tag. This is a point I found out. The other point is the description of the tag, which is missing. In GitHub, I have the tag, let's say "Code," but I don't have an explanation for it. I don't know if it depends on the customer configuration. As it is, the default configuration does not provide the description for the tag.

Palestrante 1: OK. You mean, for example, the default tags are "TD" and "Non-TD," and there is no description?

Palestrante 2: Yeah, exactly. They just say "SATD" or "No SATD," but they don't explain what.

Palestrante 1: OK.

Palestrante 2: In the menu of GitHub, you might see tags with descriptions, but the bot tags do not have a description. It should be there, I think.

Palestrante 1: Interesting.

Palestrante 2: It works.

Palestrante 1: Yeah, and this description should not be just another piece of text in front of the tag name, for example. Not "SATD-dash-something"—it's a description.

Palestrante 2: I can show, maybe sharing the screen? Should I share the screen?

Palestrante 1: Sure, sure, no worries.

Palestrante 2: OK, so… Can you see my screen?

Palestrante 1: Yes.

Palestrante 2: OK, so if I go here… and open it… OK, you see, I have a label…

Palestrante 1: Yeah.

Palestrante 2: If I go to labels here… we go. You see…

Palestrante 1: OK. Yes.

Palestrante 2: Something isn't working.

Palestrante 1: I see.

Palestrante 2: Instead, the bot tags do not have a description.

Palestrante 1: I see indeed.

Palestrante 2: But to be honest, I don't know if it's needed because it's automatic. But probably it should be helpful if someone opens the issue and wants to know what it means.

Palestrante 1: No, definitely. The more information we can provide to people using the bot, the better. So, as you said, this is relevant to understand what the tag means.

Palestrante 2: Yeah, if I open an issue and see labels like "SATD," I might not know, depending on who opens it. Maybe the developer who installed the bot knows, but someone new might not. I think it could be helpful to have a description. If I hover, I don't have it.

Palestrante 1: That's great. A very nice feature to implement.

Palestrante 2: And the last point is related to comments.

Palestrante 1: Yep.

Palestrante 2: I'm not fully convinced about persistent comments. The fact that I issued this command… it's not so relevant. In the comments of an issue, this could be boring. So, manual or automatic mode for the bot… I didn't try the automatic mode, but if I write a comment like "label," it's not interesting for someone else to see. Why should there be a persistent comment?

Palestrante 1: Yep.

Palestrante 2: At the end, I don't think it's strictly necessary to have the comment.

Palestrante 1: Something like… when you give the comment to the bot, it executes it and then deletes the comment.

Palestrante 2: Might be an option—deleting the comment, but maybe keeping a history like "bot deleted the comment."

Palestrante 1: Yep.

Palestrante 2: So at the end, you're not gaining much in terms of cleaning the dashboard.

Palestrante 1: The issue, OK.

Palestrante 2: Yeah, because in the issue you might have discussions and information, but this is just for manual mode. I don't know…

Palestrante 1: Yes.

Palestrante 2: The manual mode seems more like "try it out." In production, automatic mode should be 99% of cases.

Palestrante 1: Great.

Palestrante 2: Just an opinion.

Palestrante 1: Yeah, sure. Very valid comment. We will definitely check it.

Palestrante 1: Another question: you mentioned it's nice if the bot could check the tag you put on your issue, but also assign a label without checking the previously assigned tag. Do you think the bot helps identify technical debt quicker in your issues?

Palestrante 2: In my experience, technical debt issues are usually opened by developers. The developer knows it's a tech debt issue, even if they don't know the exact solution. They create the issue to track a shortcut they took. Going backward to identify issues is another case: you might review code written by someone else. In most cases, developers know the problem is due to technical debt.

Palestrante 1: OK.

Palestrante 2: I would say the bot is useful for preventing human errors—if a developer forgets to tag an issue, the bot solves the problem.

Palestrante 1: Yep.

Palestrante 2: If the process is straightforward, I trust the developer—the issue represents their intentions. But the bot is helpful when humans make errors.

Palestrante 1: Sure.

Palestrante 2: For planning technical debt repayment, if all issues are tagged, you can have the list in one click. Otherwise, you would need to check issue by issue, which is a huge problem.

Palestrante 1: Yep.

Palestrante 2: Searching by description is less accurate than a machine learning model. So with high numbers and more frequent errors, the bot is very useful.

Palestrante 1: Nice. So in contexts where the number of issues grows, the bot is more useful.

Palestrante 2: Yeah.

Palestrante 1: Perfect. This concludes the usefulness section. Now we move to ease of use. You already mentioned it was easy to install and learn from documentation.

Palestrante 2: Yes. Even though I did not check every configuration option, the installation and configuration were straightforward. The repository provides examples.

Palestrante 1: Yep.

Palestrante 2: It was very quick and I did not have any issues.

Palestrante 1: Great.

Palestrante 2:

Palestrante 1: You also tried different issues. Was it easy to change options in the configuration file?

Palestrante 2: I didn't really play with the configuration much. I changed my email at the beginning, then pushed it. Later I adapted how I worked, e.g., forcing the bot to classify as technical debt or not. The payload in configuration is the description, so I adapted to that.

Palestrante 1: OK. Could you find the payload type option easily?

Palestrante 2: Quite quickly. The configuration is about 30 lines. Most options are obvious.

Palestrante 1: Nice.

Palestrante 2: But this is based on limited use. I did not work with it extensively.

Palestrante 1: Sure. The goal was to get your perception.

Palestrante 1: Do you think it would be easy to configure the bot without the documentation?

Palestrante 2: Yes. I had seen the bot working before approaching documentation, so I was a bit prepared. The configuration is quite clear.

Palestrante 1: Great.

Palestrante 2: I did not face any problem.

Palestrante 1: Great. Last question: any other features to make the bot more useful?

Palestrante 2: No, I had three main points: persistence of comments in manual usage, conflicting tags with manual tags, and label descriptions.

Palestrante 1: Great. Thanks a lot, <omitido>, for your insights on evaluating the bot. Very useful hints.