Dawid Wozniak:
OK, so let's start. As you can see on the screen, it's transcription coming on. I'm not recording the screen or how we look with the camera. So, let's start with the short presentation to explain the role and then I will show you the product. And at the beginning I would just give you my research question and it is more complicated than that. But here on the slide you have “let's make a commit history great again” and it is like a short version, but the full version is - which insights about commit messages are useful for stakeholders when visualising a project using GitTruck? So, you can think about the GitTruck as a car that I'm going to show you. It needs some improvements. So, it is why we have this car on the screen, but it always goes about the commit messages. I can see now we write the commit messages and then we come back to our work, we write next commit messages, but we don't really use it for an automated process to help us with the visualisation of our projects or changes that have been done. Their only purpose is to use when you're explicitly want to go to some commit and see what's going on, like manual operation. It is a huge database that we might use. The problem is that we don't really know how to do this. So, you think about it in the terms of the traffic lights or traffic signs. You might think that OK, yeah, I have some information that I would like to give it to the developer or to the product manager about the project. But how should I give it? It is like when we want to say to a car driver that they supposed to turn right. We might do it in many different ways and conceptually we might use some colour. We might use some shape, we might use transparent or field arrow to give the right sign and to distinguish between those signs. The idea is that we want to get there, so we want to go on the full speed in the right direction and it is what you can help me with. So in this metaphor you can think about yourself as a mechanic that I'm giving my car to and you can help me tune it a little. So, then we might use this database to help developers and other stakeholders. I'll give you the tools. The tools that we have to use is my knowledge, my skills and my time to implement your suggestions and suggestions from other people. To be able to do that you need to provide me some feedback. So, you need to take the car and you need to list what should be changed improved, added or removed. Finally, we will iterate over the solution. So, I talk with many people, the testers, and at the end I would like to talk with you again and see. If your suggestions were implemented because some people said that it would work better with it or this was a suggestion that cannot be implemented because of this and that. And so on… that's the whole process and there would be like following sessions. Let's start with this one. At the beginning we start with this screen. So here we supposed to see the list of the repositories where this project was run. For now I will just analyse GitTruck with GitTruck as we have a small issue with the scalability so during the next meeting if you agree to have one, then we will analyse some product related to your work. But for now, we stay with this. It is a standard view that we have here. So, we have some information about the project. What is the project name? What is the current analysed branch? When was it analysed? What was the comment hash and how many files were analysed here? There is some feedback panel and then we have chat type. So, as you can see now we have the bubble type so it means that it's colour coded by some metrics. For now, it's a file extension and we have some colours here. So the bubbles that contain other bubbles there are folders and when the bubble is bigger than other bubble then it means that the file size is bigger. So by this view we can see that this *demo.gif* file is probably the biggest in the repository. The other metrics I don’t mention it's something like number of commits for particular file, last change, single author, top contributor and track factor that is combined number of commits and top contributors. That is not that important for now. Let's see number of commits, then the colour is different than we could see before. This file was probably the most changed. This was just two times changed. So, it is somewhere back on the scale and it's very light. Let's stay with the file extension and when I click on some bubble I get the detailed view. There is some more detailed information and finally there is that comment history part that I'm responsible for. So, for now it works very simple. We have the three commits. If there's

more, we can expand this view and it is from the latest to oldest commits. So when I have this view, I can expand the accordion and then we see the message of the comment. Here it is just some automated bot process commit. But if we have the file that needed be edited more frequently like this detail file, the list might be very long like this and when you have one day that contains more than one commit. It's just separated by the new line. So, my first question to you is, what is your general impression, first impression about this product?

Enrico:
So. There were a few metrics. The list of metrics here. So, I think that the visualisation is useful, but there are some metrics that I don't think I will ever use. For example, for me, like file size, so the fact that the bubble are larger with the file size is not very useful, meaning that I most likely don't need to know how large a file is to know whether I need to look at its history or whether I need to know something about changes in the past. But others are more useful, so like a number of commits might be useful because it tells me which file is very much likely touched for example. So, I see this as a metric. So right now, you have basically two metric two like visual clues of what's going on, which is the colour and the size. Changing the colour is very useful because then you can choose which option you prefer. Size metric might also be added if it's not there already, otherwise that might be also something interesting to do. So, like, I don't know if that could be worth it. I mean that I think… this metric list makes sense… What I'm thinking is that… If there's a better way of having those because, right now, if I'm looking for the specific metric, it's kind of hidden within the list, which is fine. Another thing, I need to click a few times to basically dig into the commits if I really want to look at them. So the date is useful, but maybe also having a way that minimises my clicks might also be useful, like that gives me an overview or something… but it self makes sense. I think it's interesting to see the visualisation of which files were changed based on folder and such.

Dawid Wozniak:

OK, great feedback. For now, I will give you some statements and you can tell me how much you agree or disagree with them. So, the first one is that they commit messages when you read or write them are important to you so it's important what is written here for you when you read or write it by yourself. The scale from one that's completely not important to you to ten that is very very important.

Enrico:
Between 7 and 8. So, I would say maybe 8.

Dawid Wozniak:
Now when we talk about UI…

Enrico:

Sorry, small note, in our particular workflow, the way we work here what I'm considering as commit message is what is finally committed to master. I'm not considering private branches because there are a lot of times I write just like super fast commits and I rarely go into someone's branches to look at these commit messages. I usually look at commit messages only in release patches.

Dawid Wozniak:

Yeah, I'm talking also about the release branches. So now, I mentioned something about UI so if you need to grade the UI in the scale from one that is completely not usable to ten that's perfect and nothing to do to improve this view of commit messages, how would you grade that?

Enrico:
I would say it's a 6 and 1/2 so it is good enough, but it's not completely immediate to search and navigate within it.

Dawid Wozniak:

That's OK. And if you have this GitTruck in the current state, how beneficial would it be to use it in your work project but also in your private projects from one that means you will never use it to ten that you will use it everyday multiple times? What is the grade of GitTruck usefulness for you?

Enrico:
I think again around between 6 and 7 so I could use it and I don't think I would use it like on an everyday basis or multiple times per day. But I can see definitely some use for it.

Dawid Wozniak:

Okay and now we finish with this one to ten scale questions. But I would like to ask what would be your main goal when you use Git track in general when you have such a view for your repository that hopefully, we will have during next meeting. What is your main purpose to use GitTruck?

Enrico:
Find a specific change that I'm not super sure where it is, but I know maybe the folder where it is so I can have like a quick look or jumping from one side to the other side when it's changed and like what change it is. So there could be used for statistics or like… I think in the metric you also had the person who is a last commiter or something or… it's not last commuter but it's about last change. So, it is also by date. So, if you want to do this okay, okay… that wrong.

Dawid Wozniak:

let's change it so going above here we can see when was the date when this file was last time. There is another metric that is called single author. Yeah. So then it means that this file was created just by one person and then you have top contributor. That is the person that contributed the most to the particular file.

Enrico:

Yeah. So that that could also be the same user, right? So you. But again, if I'm looking for a specific change or a specific area then figuring out who is the top contributor… it maybe be top contributor to that file that knows it. It is something I could use. Because if you look at it, you can get it. Also if you look directly in the commit history from it…

Dawid Wozniak:

of course…

Enrico:

But then having it visualised gives you more like, for example, who is right the main contributor of a file, a certain area in the product that has multiple files. So, this view is good for that as well I think it's easy to jump from one fact to the other here. If you have a history from one file, you can always look it up but being able to have the overview of the folder that's good.

Dawid Wozniak:

Yeah. And if you think just about the commit history, so for now, we have just the date and the message, but we have the access to the all information from the commit. So, is there anything missing that you would like to see in this view?

Enrico:

I don't think there is missing information. Maybe the person that did the commit which is not here and it is not easily accessible but the main thing I would say it is what I would like to have… maybe it should be more structured. So, like for example if you open one that was, I think history where there were a lot of commits. Yeah. Here for example. It's not immediately visible to me how many commits are there, like how one finishes and the next one starts. Also, sometimes you have very long list. I don't know how you handle it, but if you have a very long commit messages I want mostly an overview, maybe. So, I want to see like how many there are in that particular date and to get the glance of what they are about. It means like I don't think a lot of information is missing, but I think UI wise it should be more visually clear and who is the committing person is.

Dawid Wozniak:
OK, that's very good feedback. Now, I have more, maybe, abstract question, but it's also related to this commits study. So, let's imagine that you have like unlimited time, unlimited resources, unlimited money to complete some ideas, then what would be the most useful in insights from comimits, especially commit messages that you would like to have in this project visualisation.

Enrico:

Good question. I'll tell you one thing that I always do when I make commit messages. I like the ones that have, at least in various branches, the first word tag like in square brackets with the area that that that commit touches. I always put it there. For example, I could put “Power BI” or tag “automate”. Because if I, for example, just see something that is “admission hide icon”. Where is the icon? Normally, where does it show up? Which part of the product is that changing? I don't know how it is possible to take out of commit messages, because, of course, people don't always do nice commit messages.

Dawid Wozniak:

Yeah, but we have access to all information about the comments, so we might just add some extra information on the UI, so it's completely valid answer.

Enrico:

So that's is my, at least one thing from commit messages that I miss. And then… maybe you change this file as part of a big commit… which are other more likely files to change when you change icons, for example. Did you change these file just because, I don't know, you fixed the comma or reference or like did you change this file that is referencing differently to a new icon or this file is actually containing the icon. I don't know, I feel like the area… the area that the committer should change probably changing this file. I don't know how to define area, but... It's very depending on the repository.

Dawid Wozniak:

I think that is right and clear. Is there anything else that you would like to tell me about the product that I haven't asked about? Some suggestion, comments or possible improvements.

Enrico:
I have a question actually that I had from earlier stage. So the size of the bubbles. Is that a logarithmic size? Is it? Cause what I'm wondering is this bubble in bubbles are very much larger and I don't think they ever need to take that much real space so that that's for the visualisation… and the committer part, I think I already told you that, the date is there is good but maybe a way of sorting not only by date, but also the committer. Access without having to click to expand in every row is the other thing that would be probably useful. Maybe something like a focus mode on the commits, kind of thing.

Dawid Wozniak:

So, to answer your question about this bubble so the minimal size is that we have the minimal file size - we take this for the smallest file in the repository. I think for this case it's this one and then it's the ratio between this file and the biggest file. That is why it's so huge, because this file is probably like 1 MB. Yeah, and it's like 58 bytes. And here you have a megabyte. So, if you calculate the ratio, it's a big change. I mean it is a good suggestion that maybe it's a good to have some way to place the bubble in the different way…. Okay so it was my last question about this particular product. Now I will like to ask some statistical questions. So I have feedback from very different group of people and then I would like to do some statistical analysis. So I need to ask for some information from my testers. My first question is how many years have you worked in the IT industry including part time jobs. So it doesn't need to be very precise but number of years.

Enrico:
Six.

Dawid Wozniak:

OK and what was your primary role during this time?

Enrico:

always software engineering.

Dawid Wozniak:

In average how big was your team including all so developers QAs, product managers, product owners, but not the management, just people involving in the one particular team managing some projects, yeah.

Enrico:
Between six, seven and maybe ten or eleven. So average I would say eight.

Dawid Wozniak:
Okay and when you have your average project, how many commits are there? You can say per day per week, per month, per year, what is more comfortable to you? And I'm talking about coming to master, not in the repository private branches.

Enrico:

Yeah. Yeah. So I would say maybe 10 per day.

Dawid Wozniak:
10 per day. That's a good answer. And how many files are in your average project? So also we

calculate only files that are meaningful, so not like npm modules or like configuration files for Visual Studio code or other editor. So how many files are in average there?

Enrico:

Between 10,000 and probably yeah, 100,000 depending on the work.

Dawid Wozniak:
OK, that's good. And now finally, I have the comment about the next iteration. So you can think about this as a version 001 and then I will implement some feedback. And I would like to present it to you again. I will address things that you said to me. And I also tell you what other people said and so why I implemented this and not that how it's going in which direction etc. At the end there will be a special survey. When we have version of GitTruck with the new final changes and without them, we will try to compare and see what the benefit for you is when you use this second version.