

**Dawid Wozniak:**

Hello, Monica. Thank you for participating. As you can see on the screen, now I'm recording this meeting as a transcription and then I will send you the final results as probably it is not always correct, but we will start with this short presentation then I will show you the product. So, let's do that. So, my question that I'm trying to answer is different and I will tell you now but in general, what I'm trying to achieve it is to make the history of commits great again. My whole research question is which insights about commit messages are useful for stakeholders when visualising a project using Git track. So, you can think about the GitTruck as a car and then we need to do some maintenance. So, when I have the commit messages, I think that we write them and then we just turn around and we come back to our normal work what we need to do. If you think about it in the terms of the traffic, then it would be like we go there and we turn around and going into different direction. So, we want to get something more from the commit messages, but we don't know how exactly we want to do that. It's similar to the traffic analogy when we have to give some signal to the car drivers. But there are many ways to do that. So, if you want to say that you should turn right, you can have different shapes, different colours. What should be colour? What should we? How you distinguish between those four signs and their meaning. So, it's that kind of problem. But of course, we want to get there. We want to have the right signs to achieve. It's what we want to achieve in the project. So, you're gonna help me to go there. Imagine yourself as a mechanic that I'm just asking for some advice about my car and then I will give you the tool to fix it. So, the tools are my skills and whatever is needed to fulfil your wishes. But you need to provide the feedback so you have a car and then we need to talk what actually can be improved and how. In general, what do you think about it? Then as we have many people and many suggestions, we will iterate over the solution to get the best results. So, it is the end of presentation and now I can show you the product. So at the beginning you have just the list of repository for this particular demonstration, I will analyse GitTruck with GitTruck itself. It is because we have a small issue with the scalability. So, if you need see a big product, it'll take forever to load, but hopefully I will be able to fix that. So, we have here on the left side some information about the repository in general like the title of the report, the branch, when it was analysed, commit hash and how many files are in total here. Then something to provide the feedback and then the types of chat. So, as you can see on the right side there's a bubble chart now, but you can change it to the tree map and it is something similar to the old file explorers, so let's stay with the bubble one and we have some metrics. So now each colour corresponds to some file extensions. You can see that the blue corresponds to TypeScript files and some other colours to different types. We can have also other metrics like number of commits. It means how many times the file was changed. Last change, single author, top contributor and the track factor that is actually combined number of commits and top contributors. Then let's come back to this file extension as it is more clear. Then when I click on some particular file. So, if the bubble is bigger then it means that the file itself is bigger than others. So you can see that this is probably the biggest file in the repository and then we have some smaller files. You can have also bubbles that contain other bubbles. They are just folders. So when I click on something then I get the name, the size, how many time people change this particular file, when the last change was and the full location, author distribution. So here, it has some bots that just updated something and here there's Jonas. Then, you have a commit history that's actually the part that I'm taking care of. So at the beginning it's very similar to the author distribution. It is very simple. We have by default 3 latest comments and when I expand this view. It is the date and then I can see what the message was here and here. We have also some other commits. So we have two more and I can also see them. If we go to the file that is more edited like *details* component, then there is a lot of commits coming on here and you might have a long list of dates displayed like this. When you have one particular day like a second May here when you have multiple commits then they are just separated by new line. It means that there were

more than one commit at this particular day. So, my first question is about your first impression. What do you think about it?

**Monica:**

Ohh I already have many recommendations but I'm not sure how much... so you're only taking care of commit history and you're saying all of the tool is kind of already available. I ask because I haven't used this tool.

**Dawid Wozniak:**

Yeah, so this tool is available but without this coming history feature.

**Monica:**

Ohh I see okay. So first and foremost, right, it's very good that we're having this meeting now because I actually had a problem yesterday, which is very relevant to this project, which is where we have a process where we do send the strings in our project to another team. They give us the translations back and then we insert the translation again, right? So there's a translation team that is only taking care of the linguistics part, you know. So let's say the string there is an error message, they will translate it in 10 different languages. They will give me the json file and I have to insert it. So my problem yesterday was I wanted to figure out when the last translation was made in the project. So, if we take that example, right? First and foremost, I would like to have more, I know, it's not part of commits... completely directly related, but would it be possible to have more than one metric in the chart? So what if, I want to see json files that are, I don't know, created by Dawid...

**Dawid Wozniak:**

Yeah, so it's not possible now, but it's relevant for this project as it is the part of the commit in general.

**Monica:**

Exactly. So if I for example know that these are the three developers that are responsible for, let's take this example, that are responsible for inserting translations. So, I want to have, first of all, the filters. You know, give me all the json files in this project that have been ever committed by these three developers, right? And then I want to see if the commit message contains certain keywords, for example „translations“, „language strings“ or something like that. So, I think that's where commit history message could play an important role. Like that or either you can add it as a metric itself where you filter by some keywords. I think you have to take care of in it in a way that it's logical and or maybe also provide logical description of how you wanna take care of it. It's the same way we search in deep or something like that. So that would be what I think I would love to use. Another thing that I'm very curious of why are we only showing 3 common messages? Why is it?

**Dawid Wozniak:**

So the problem was that when you show all of them, and you want to scroll actually to do some actions like you can hide this particular extension or this file because you don't want to look at it. You have a very long list in the project when you have over 100 commits then it's like scrolling and scrolling. So, at the beginning it was said, okay, let's show three and we will see how other people react if they say - can I have more? Do they want to see all of them? Or maybe it should be separate panel like on the left side we have 3 separated panels and here it is just one.

**Monica:**

Maybe, it should be another tab. Yeah, I think you need to think about the UI. My first reaction is - I

do wanna see more than three commits. Usually it's because we're, especially, we have bigger projects and people checked in a lot even in one day. You can have more than 30 commits on a file or something like that. So, if I want to search, my main scenario, when it comes to commit history, what I want to get, is usually... I want to find what regressed. Which commit regressed? If I know there is a bug in it or... when the last time this was touched and that's why I usually look into commit history. So, if you take these two things into mind and then design features around it, I think it will be very helpful for any developer.

**Dawid Wozniak:**

Yeah. That's good. So you talk a little about the UI, if you look have a look at this product in general and it is the one point and then another just for the comment history, how would you grade the UI from one, it means it's very, very bad to ten that it's excellent and nothing to correct actually here...

**Monica:**

I am not a UI expert, but just from the user point of view, I would say that three commits are not enough. Maybe, three commits are enough in the beginning and then I would like to drill down some panel shows up and I can be completely focused on commit history for the certain file. For the UI of general, I feel like this metric showing just the file extension is not itself sufficient. I would want to have an ability to add more metrics. I know it's not like more UI. It's more feature requests but apart from that, yeah bubble charts work for me. Maybe more chart types could be also helpful for other users. Maybe, list... I think the chart that you're showing me right now helps when you are trying to find out, OK, I want to change certain feature in the product. How many files do I want to change? Is an answer possible with this chart? How many? So, let's say if I want to change, for example, logical separation of the product, right? Maybe this chart will give me better answers saying OK if you want to touch this area of the product there might be 10 files that you have to work with which is some good answer but I'm not sure how much we will actually use it.

**Dawid Wozniak:**

okay and but in the scale, for like giving grades from one to 10, how would it be?

**Monica:**

Yeah, I didn't really answer your question. I see. OK. Ohh, I will give it six for both commit history and for the project in general. Wait, six to commit history... as it needs to improve, so I'll go a bit down and give it a five.

**Dawid Wozniak:**

OK, that's great. Now we can talk about the commit messages itself. So, when you have a commit message as a developer, you write them, but you also read them. So how important for you is what do you read, what you read and what do you write there from one to ten? How much attention do you put into writing this?

**Monica:**

We actually, at least I personally tried to write as meaningful comment history as possible, because I'm also a user on the other end, right? I read other people's commit history messages and sometimes the messages, they are committing like "feature" that is not useful, I would never know what's happening in that from it. So, then I have to drill down and read the code to figure out what happened in that comment. So, for me it's very important. I would say 9.

**Dawid Wozniak:**

OK, that's great. And the last question in the scale from one to 10, if you have this product now in the shape as it is there, would it be beneficial for your work and for your projects? It doesn't need to be like work related if you have personal product, it can also apply to them. So, scale one to ten, what would it be? Not useful at all, so one or ten, very very useful.

**Monica:**

Ohh so to be honest I wouldn't use it if it is outside my repo - my Azure develops but if it is integrated with Azure DevOps. I will use it. It will be useful and good. I don't know. So, I'm gonna say... personally, me, I would use it, every time I will do anything with commits. So let me say, if it's integrated as it should then I'll use it. For now, eight on a scale of 1 to 10.

**Dawid Wozniak:**

OK, that's great. It was the final question in the scale style, so now there will be no any scales, but when you have this product you mentioned that your main goal to use it is to find out when was the last commit, what caused the regression. Did I get it right?

**Monica:**

Right.

**Dawid Wozniak:**

So another thing that I would like to know, if you have like unlimited resources, time and everything around the project, what feature would you add to commit or commit messages history to get some insights. I mean What would be the most useful insights about commits, especially commit messages? Do you have an idea?

**Monica:**

As I said, I would really like, so I ideally I would love to ask a question like give me an answer saying "when were the last five commits related to this feature?", "which change caused this issue?". This is what I usually look for when I look for commit messages, right? For example, I told you, one to find when the bug was, when the feature regressed or another which is like I want to do some more file changes, but I want to do who did the last time change, right? So, I will look into the history and I also look for commit messages, maybe, even to take inspiration from the code of others which have done the same thing. So that is a good to look into commits.

**Dawid Wozniak:**

OK, that's perfect valid answer. And is there anything else that you would like to add about GitTruck or anything related to this project, not only to the commit history, some suggestions?

**Monica:**

So, if you're really focused on commit history, first and foremost, my suggestion is to show it. From UI point of view, it's really hidden. So, bring it, bring it in a way that it is noticeable. I can understand that you have the constraint of the size, but at least have a drill down option so it can open in a full screen mode when certain person knows that this is the file for which I want to look into commit history. I know this is a feature that is already available in lot of other git tools, but I think what you have here already. It makes sense to not go to another tool or another page just to look into commit history. So I would say that first and foremost suggestion is to allow for us to read them because we also add images to commit messages, right? So, the commit messages could also be very big. There is a team that that sits in Microsoft and I know one of the developers there. Honestly, he writes

about literally every line in the commit message, which may or may not be a good practise. We could talk about that later, but I'm saying description of commit messages can get very long. So this much UI dedicated for it is really a hurt for me personally.

**Dawid Wozniak:**

OK, so it was my last question about the project itself, but now because it is our first meeting, I would like to ask you some statistical question to see how the feedback differs based on many factors that are just about you. So how many years have you work in the IT industry more or less, including part time jobs?

**Monica:**

Ohh that is 6 + 3 + 2. It's almost 11 years now.

**Dawid Wozniak:**

OK great. And your primary role has been...

**Monica:**

software engineer

**Dawid Wozniak:**

That's great. And during your career you probably were in many teams. So, what was the average number of people in the team including developers, QAs, product managers, product owners?

**Monica:**

I would say 6.

**Dawid Wozniak:**

OK. That's a good number. And you probably also participated in many projects. So, when you have an average project and you have a group of people committing there, how many commits are there per day or per month or per year?

**Monica:**

I don't know the answer to that question. I really don't know. I never counted number of commits.

**Dawid Wozniak:**

OK. And then the number of files...

**Monica:**

I mean, we have very big projects. If I have to give an estimate, it's definitely around 5000 or even more.

**Dawid Wozniak:**

OK, so that was the last question. So, as I said at the beginning, it is the process we want to iterate over the solution. So, you can think about it as a beta version one and then I will create the beta version two including feedback from you and other testers. And then I would like to show it to you. So, I will send the invitation for next meeting. If you agree, then I will not ask those statistical questions again. I will just focus on what improved and what you think about the improvement.