Blazej:
You're recording, right?

Dawid Wozniak:

It's not recording, it's just doing the transcription. So, we're not recording the screen. I'm just having a transcription. So, then I just have some document that it is what you said it is not what I just made up here. I think that is good and you don't need to write it by yourself. We'll just have the conclusions from this meeting. So, let's start with the short presentation. I will talk about the general idea. So, this is about the commit messages in general. My research question is which insights about commit messages are useful for stakeholders when visualising a project using GitTrack. So, in short, let's make a comment history great again. Imagine it as our product is a car and we need to tune it a little. So, for now, we have the commit history, but we don't really use it. So, if I think about it in the terms of the traffic, then we just have something we put it there and then we immediately turn around and we go in the different direction. We don't use it anymore or we don't use it as often as we could. The problem is that we don't really know what is important. What is the insight that we might want to have, so when you have some situation that you should turn right, you can present in the different ways, and you need to see which way is the best in the given scenario. It is what we want to figure it out. We want to get there, and we want to go on the full speed on this car and. You can help me get there. So, you can imagine yourself as a mechanic for this car and you are just help me to tune it a little. I will give you my tools. So, I will give you my skills. Everything what we can do but you need to provide the feedback. So you need to take this car and say you should change this you should probably add this or we don't need it anymore. At the end we will reiterate over the solutions. There are some suggestions. We need to decide what is the most important thing, what is slightly less important that what is just nice to have but it's not that urgent. So let's start with the presentation of the product. So here it is just the GitTruck first view. So, we have one project GitTruck, it's just the visualization of this project. So, for now we have a small issue with the scalability and because of that I just show how this project is built. We have this view. So, on the left side you have the title of the project, the main branch, when it was analyzed, what is the last commit here, and how many files are in general here. It's some feedback box. It's not important. There are some settings for the visualization of the right. So how you actually want to see this. When I click on one particular file, let's say this one, it is some *demo* image. Then I see the size, how many commits were changing this file, when the last change was, what the localization is, some author distribution. So, it means that out of this 8 commits 88% was done by some bot „VHS action" and one was done by real person Jonas. Below, you have a commit history. That's the part that I am taking care of. Here you have a very small view where at the beginning you have just three latest commits. You can expand it as I do it now. Then you have some dates and you can see, OK, here we just add this file. Here at the end there were some updates for this file made by bot. So maybe some bot optimization is not that interesting, but if we go to file *details*, then we see that there are so many commits. That's where the list might be very long and also for some days let's say second March. There is more than one change, so you see the whole list. It is just the new line that's separating this list. It's time for your first impression if you want to express something how you feel about it. What is your first impression when I've shown it to you?

Blazej:
I just wanted to, you know, understand what I'm saying. But as I understand, this is the like a structure of the of the project, right? And the individual bubbles represent specific component or file. And then you have the commit history. Why? Why is it structured like this? How are you

deriving the sizes of those components? Or you just let them do it. Is it important for the sake of the project or not?

Dawid Wozniak:
It's important that you understand that. Fortunately, this is actually the structure of the project and how do we create this? So, as you noticed, we have some bubbles that actually are files. You can hover over it, you see the name of the file and then you have some big bubbles that contain other bubbles. They are just folders, and the size of the bubble corresponds to the size of the real file. So here you can see that this big file is *demo.gif*. So, they are the biggest in the repository and there are also very small files like here we have some file that is very small. So, if I click here then I zoom in just to this folder and there is a path to this just *husky* and pre-commit statement that you want to verify before you commit to this repository. You can also zoom out. So the idea was that here you have the overview of the structure of the project without going into the explorer and exploring all the things. You can also notice which file is the biggest or the smallest just by looking at it, right? When you click on the specific fie, you just click on the *detail* component. Then you probably want to know more so we have the detailed view on the right side, and you have some hidden files that are by default here. Json and PDF so, we don't want to show it in this project, but you might edit this list. I want to see this and then you have some extra information among this extra information. We have also the commit history. So, this extra information should help you to understand the project. The general purpose of this is to imagine that you are onboarded on the new exciting project that was developed. It's now in the demo state. So, you have some feedback to provide but you also need to understand what's going on to see where actually are areas that are assigned to specific people and you don't have that much time to talk with them. So, you need to figure it out somehow by yourself or you want onboard another person to your project, and you say, OK, so if you did something here then you have this as well. So, you probably ask team owners or Tomas, and then we have this commit history with some description of the changes that were made to this file. **H**ere it is rather small description of the changes. It also varies from project to project. In our project, we usually have a very long description of the change that we made and what was the motivation, the project focused on commit in general etc. So how can I make this commit history more useful at all? Can we get anything from those messages? **Is** it useful, or maybe some other properties of the commit are more important? The commit contains all the properties of the particular change. So, when you have something like „update repo" and „migrate import statement". It means that here we update some framework, so we have message content and changed files. So, we have all these information. We have a date, we have who committed this, when it was committed, we have this message that was defined by a developer and we want to have some insights that will help to visualize this project.

Blazej:

Right, so far, I've understood that what you're showing is like an alternative way to browse through the project and through individual commits. Is there anything like the next step? When you're analysing those commits somehow or no possible?

Dawid Wozniak:

That's the thing we would like to analyse it, but the thing is, what we want to achieve, what we want to get… We don't know. So, if you think that it'd be good to analyse them, what would be the outcome that you would like to see from this analysis?

Blazej:

That's a good question, I think. Yeah,

Dawid Wozniak:

This is something that I'd like to understand from you.

Blazej:

You are trying to figure out the purpose of the project, right?

Dawid Wozniak:

Yeah,

Blazej:

The visualization of the project, graphical visualization with bubbles, that's very nice. It's cool but it doesn't give you anything on top of the nice navigation through it. This is interesting. It looks interesting but it doesn't have anything to do with commit history. So far, for instance you might want to try to visualize, you know, what changed, when and whether, if you're looking at this visualization. It should be like you're looking at commit history. So, for instance, colour code somehow, you know, change to files that were there changed in the last week. There are some files that were not touched for long time and so they could be like differently coloured, or I don't know that might be some visual glow in terms of like the changes happened to them. Where was the commit history most active, which means basically you are there where people are editing stuff at this phase of the project, right? So, I think in this graphical representation that might be interesting because he will see where the action is.

Dawid Wozniak:

Yeah. So, for now, we have the file extension. It is the colour. So, it means that all blue files are typescript, but we also have other metrics. Now, we have file extension. We have also number of commits changed, last change. Let's say, last change and here you can see that if it's darker blue it means that it was edited recently and if it's very light then it means that it was a long time ago. So, you have already some metric uses for this information. We also have something like the number of commits. So, it means how many times, files were changed how frequently, so here it seems like this file was changed very often comparing to other files. Yeah. We have also single author. So, it means that only one person committed to this file. And here you have just simply yes or no.

Blazej:

So, this is now interesting. Right. So, you're changing the chart type and metrics.

Dawid Wozniak:

Yeah, you actually see what is going on, yeah, so the chart type is bubble or if you have more complicated structure than you might have this problem that some bubble are very very small. So, you might prefer to use this tree view. This tree view looks a little like in some old file explorers. So, then you have a big square that is split into some small squares. It's clearer for some people, but let's stay with the bubble and then we have top contributor and track factor. That is actually very similar to top contributors. So here, we have the person that committed the most to the file. It's also color-coded. Finally, we have the truck factor. It is similar to how many people committed, so here you can see that there is just one author and here 2 authors. But if you go to the single author or top contributor, you can ,kind of, derive this metric.

Blazej:

Now I understand that better by looking at the different metric. You can actually look at the commit history in a visual way so that looks nice. Is there a way to do some filtering or searching based on some keywords or maybe some other things… When you said make commit history great again, visual representation is one thing, but another what I thought… what came to my mind is something more about… you know, sentiment analysis or text analysis, right? We can see and we can show like what type of commits happen or I would like to see only the commits that tell us something about bug fixing. Right? And then and some other would be… I don't know. I think mayhbe the latest version of a component… or some... list of keywords and then yoou filter by those and limit the visual link to them. The visualization would then change. So, for instance, I would say, show me the the files that changed the most, but only based on that particular keyword, then it would be nicer, I think. You can also some somehow maybe filter out ones that are not needed. In this example, you have pretty much like a good distribution, right? But in in the larger project you might have some files that have like 1 commit, because someone created them, added, and that's it. No one touched them forever. So, they are not interesting for the analysis. You can kind of hide them. Instead of showing like a lot of those little dots like this *Husk*y, those *Husky* files, right? This is just some config file that someone created.

Dawid Wozniak:
If you think about those categories that we might classify based on the commit messages. What would be the categories you mentioned?

Blazej:

You know, you would have to analyse the text, so it should be dynamically based on the project. I think because every project might have slightly different content. I mean you can have some form of… even AI, but like some form of text analysis and say that, like, you know, quite often you will have some predefined keywords like bug, uptake, component or whatever. By the way, you would like to have popular ones listed 1st and then anyone can either add their own keywords or you actually search through the commit history and figure out what keywords are. I use the most right. Probably there is some way to do it in a smart way. Remove all words that you have in all commits like ending or into and stuff like that. So probably there is some easy way to do that or something that does it for you. Maybe create this kind of cloud map what keywords are used the most in given repository and so on, sentiment analysis, that type of a thing, that could be something good. You wouldn't see this in the commit history. This is your key. These are your keywords. Then you can kind of click around and see how the visualization changes based on those fields.

Dawid Wozniak:

So basically, filtering, right?

Blazej:

Yeah, alright, I think it might be good. And I'm guessing if you zoom in, you would click on components only, right?

Dawid Wozniak:

It would zoom in, yeah. You will see only the stuff that is in this folder.

Blazej:

Yeah, it makes it easy for me to kind of navigate and see what changed, when, the same with authors. So, for instance, if I would like to only look at our changes that I have done or one of my colleagues. I'm looking for some something, he might introduce or contribute to. Good.

Dawid Wozniak:
OK, so I would like to ask now to what extent you agree in the scale from one to ten. So, ten means that you're completely agree that you the statement is true. One means that is completely not true. The statement is: commit messages are important to you. So, when you have something like commit messages. Is it important for you to what is written here or not really?

Blazej:
Yeah, I think it depends on this committer how they write those messages like you can only do some sort of statistical analysis like how many changes were for a given file, but you don't know what changed if they are bad. The most important is to be able to analyse, you know, what has happened. Well, yeah, if you want to have a in a scale to ten, probably like 9 or something like that.

Dawid Wozniak:
And so let's talk about the UI. How would you grade it from scale one to ten? The UI overall, it's one thing and then another, it's just a commit messages part here. So far, we now have just the dates from the latest commit to the oldest. So here when you have many commits then we split it in just adding the new line. How would you say? Is that good overall?

Blazej:

It looks really nice. The UI is nice, kind of fluid, and kind of dynamics. So, I'll give it nine for the overall product. Well, maybe 7 for commits if you need a number but that's only because I actually would prefer to have those parts of the commits from the committer shown on that kind of list, rather than the date, because if you're looking at the date, you can also just display it very short, you know, like 28th February at the left or at the right side. Because right now you have like space and then show a little committer information. So, like even a few words, then you will see like the committer. It will be like actually this is what was happening and then they said that because now you have just the dates and you have to open each one to actually see what was happening. So maybe you can have some form of an indication if you have multiple commits. Then you can have like either a number or some colour coding that is there. If there is more than one in the line, you can be teasing what's inside, right? Rather than just, you know, show all, because right now I have to click on each of them to find out what was happening with this file.

Dawid Wozniak:
OK, that's good feedback. And the last question from the scale one to ten. Would it be beneficial for use GitTruck for any purposes in your products, in your work?

Blazej:

You know this is a tricky question because I'm not that close to development but I think it will be quite beneficial even for me to to have a look at that. If we have a larger project, so I don't know, probably about 8. If you're talking about that theoretical example now.

Dawid Wozniak:

We ended up with this scale question, can you express your feelings about the main goal to use GitTruck? When you have this tool now in the current state and what would be the main goal that you would use this tool for?

Blazej:
I mean that probably I would be able to use it to kind of filter out visually where the changes were happening the most, because that looks like very nice. The visualization, you know, see the list of commits per file, it is on GitHub already. There is nothing special about this, right? And even you can search and do magic in that like a textual form. But that kind of visualization I haven't seen that before, so that's cool. That shows me statistics about what was changed on the repositories where the changes are. I think I would like to include more and connect that with like content of the commit history into this visualization, right? So, it's not only the number of commits, but also the types of commits and maybe... I don't know… the size of changes. Right? You can think about a lot of things that you can expose in this UI and allow me to do some additional filtering and see this map of my project. In that context, it'll be even more powerful.

Dawid Wozniak:

So, is there anything related to the commit history that you haven't mentioned yet, but you would like to have in the product.

Blazej:
I don't think there's. I've already mentioned a lot.

Dawid Wozniak:

OK, that's good.

Blazej:

Content, size, dates and what you have already.

Dawid Wozniak:

If you have, like unlimited resources, money and that kind of stuff, what would be the most useful insights from the commit messages?

So, if we can do basically everything with them, what would you like to have from all commit messages for particular repository, file or global files? Is there anything specific?

Blazej:
I mean. I think the purpose of this project is to allow participant of this project to discover its stuff that is not easy discoverable in normal ways. Uh, so yeah, like the simple things about the number of files and stuff is boring. Nothing special but finding out something strange or something unusual, or maybe something if you have multiple projects, then you will analyse multiple projects and compare them to see whether there are some trends for larger companies that would be interesting, I guess so. More that kind of analysis and statistics you can get here the better.

Dawid Wozniak:
Is there anything else what you would like to add that I didn't ask?

Blazej:

I think we covered everything, right.

Dawid Wozniak:

So, it's been our first meeting. So, I need to ask some statistical questions, I know the answers for most of them, but I need some help to get them documented. So how many years have you worked

in the IT industry including the part time jobs if you have them? Approximately, you don't also be like precise to days or weeks.

Blazej:
30 years, probably 30 years or something.

Dawid Wozniak:

That's impressive and your primary role is a product manager, right?

Blazej:

Yes.

Dawid Wozniak:

When you have the average team that you work within during your career, how many people were involved like developers, QAs, product managers, product owners?

Blazej:

Yeah, probably like 8 to 9, let’s say nine.

Dawid Wozniak:
And if you have a project, how many commits are there? If you have that kind of knowledge, you can say per day or per year, it doesn't matter. Just give me the the metrics and it's also related to files. So how many files are in the repository?

Blazej:
I don't know. I would have to guestimates, but I would probably say that usually you would have… I don't know... Maybe… 1000 commits a day from a given team. If you're talking about a small team, people are working on different things and usually they try to commit at the end of the day, so that was should be the rule. The number of files that I have no idea… I cannot even guess, yeah, probably not. Yeah, but, as you know, one thing, in here it changes a lot because there is a lot overhead than like paperwork or conferences. So, you don't commit that often because you have to have proof of concept, code reviews, and other stuff. In smaller companies that is also different, and you have more commit So, I'm thinking here it won't be that often, I guess because I mean life.

Dawid Wozniak:

That was the last question. So thank you for showing up. As I mentioned during the presentation, it is like the iteration over the best possible solution to improve this product. So probably there will be another meeting. Obviously, I’ll send invitation. So, you are more than welcome to participate. Then of course, I will not ask those statistical questions because you’ve already answered them as we were just talking about it. We’ll talk what other people said and how I implemented that, how I implemented your feedback and what you think about changes. So, I expect that it would be also interesting to see how the project procedure. So, thank you one more time. Thanks a lot and see you again.