Dawid Wozniak:
OK, so you can see it on the screen and that is basically recording what you're saying and then after the meeting, I will send it to you. So, I can start with the presentation. The presentation is about this tool. So, this tool is called GitTruck and this tool is designed to help you visualise some repositories. So, at the beginning when you run it, you will see the list of the repositories that you have here. And if you preanalyze them, then they have like pre analysed budge. Otherwise, you can analyse them. If the repository is huge, it will take a lot of time to actually visualise that. That is why I preanalyses some of the repositories that we work on. So now it's basically starting from the memory. There is still a problem with the scalability. That is why we took only 5000 commits. So, in this project, I think that is over 1,000,000. So, you one million and everything needs to go to the memory. So all information about the commits and then you have this problem that sometime it would just run out of memory. If you run it without any limits, then it works. But you need to wait like one day to know. Yeah. So, when you open this to another repository, you'll get something like this. So here you have some general information like the title of the respository, the branch that you are analysing, when it was analysed, the short version of the commit hash, that's on the top of the list, and how many files are there, the feedback box and then we have the chart type. So here you have a bubble chart. So, it means that each bubble corresponds to one file or one folder. The size of the bubble is meaningful. So here we can see that probably this bubble is the biggest one and then we have another bubbles that are bigger than others. So, if you take the ratio between the size of this *app* and this let's say *test* folder then this ratio should be more or less affected here. So, you just have twice size in the File Explorer. This bubble is supposed to be also always twice bigger than this purple here. OK, we have some metrics. S, we zoom in to some file, let's say to some folder, let's say this one, OK, still not enough. So maybe there you actually see some colours and this colour brings some metrics. So, some additional information that we would like to have here and the default one is the file extension. So we have all file extension and then we colour code them somehow and so when you hover over the single bubble you can see what is the name and the file extension. We have other metrics, one is the number of commits, then you can see which file where more frequently change. So here if it's darker, then it means that it's basically has some more commits that it's being frequently edited. The problem here is that some files were not edited in the last 5000 commits, that is why they don't have this information. Here you can see also the information about the commits. Now when you hover over it you have last change so you can see OK, this file was probably last time edited like almost a year ago. And this file is like seven days ago because it is like lighter than we have single author. It is the metric when we have basically two states like single or multiple authors and no authors in this context, we don't have the commits there. This file was maybe renamed, maybe moved. Maybe it was created for some feature and the only this person committed to this file. We have the top contributor, so it's similar metrics, but then we take all the commits and we see, OK, 6 of 10 commits were done by some person. So, this person should be the owner of this file and here for example, we have the same person because it is the one person that committed to this file. But basically, like most of the files here somehow have some owner that is also colour coding here. And then we have tracked factor. It's very simple metrics. So how many people actually touched something to this file at some stage. Yeah. So here you can see like 3 authors. Maybe one person was changing one line but we still reflect it here. I stay with the file extension and then we will go to the root. So, it needs some time to regenerate. As you can see, it is kind of crowded because repositories are huge. So, I will also disable the animation and I should have done it before. And then you have the chart depth. So, the chart depth basically allows you to make it a little faster and make it cleaner, so then you can see just one level, so it is basically what you would see if you go to the File Explorer and then you can extend it to two so you have some other layer, then three and four and so on. So, then you can also see how complicated your product is. And here on the Level 3,

you have like no files, you have only the structure. So, it means that the structure is at least four when you have some real files. So, it's also note that meaningful files are on the fourth level. I'll stay with level four then it works a little faster. So that's all I told you, it used to be there. We can go now to details of some bubbles. So here we have this app folder. Yeah. And I click on it I get some information. So it is a folder. So, I have the information about number of files here, number of folders where it is located. We have also the information of author distribution so you can see all people that contributed here at some point. Maybe I just copied one line. I would still be mentioned here, and you have some like very basic information about the commit history. So, they are grouped by day, and you can, just let's say, collapse and expand this view. So, it is just informative. You cannot click on the commit. You can just have a glimpse, but there is a special tab that actually I'm responsible for and it is connected with my master thesis. So here we can do more. The first thing is that by default they are started from latest to oldest. You can change that. So, if you are searching like okay, what was the first comment that actually I can reach here. You can easily do it by clicking this button. Then maybe you would like to sort it in different groups so you can sort it by author so you can see, OK, here we have Marco and Marco did all these commits, but I'm not interested in that. So, I would just collapse this and then we have another person. So, I also collapse that and yeah. Can just basically look how many commits are there visually, how they are distributed. I stick with the date. I think that's better for the rest of the presentation. We have also some filters. Yeah. So maybe you remember that in the commit message, you fix somewhere and it was important, so you just put „fix" here and then you have only commits that mentioned „fix" in any form in the commit message. If you are not satisfied with these results, you can say, OK, maybe it was actually in the commit description. So in our context, it is the body of this pull request, and then you have some commits like here that do not mention this word in the title but in description. So now, I can click on it and I can see OK, this is the message and here we have some description and this „fix" is here. Yeah, so here we list commits as you can see in the description, we don't support any formatting and the links any external references it is because of the security reason except the formatting. That is just the time limitation. Then we have the hash, if you want to cherry pick, exact date when was created, by whom it was created, and you can click „show edited files". Then we are fetching which files were changed with this commit and we highlight them in the repository, so in the visualisation. So, everything else should be greyed out and only those files should be highlighted here. When you have a lot of files, it might be a little problematic to find those files that were highlighted. This is why we will also get the list of these files there. So, let's wait a second, yeah. So here, it was just one this file changed, and it's still present in the repository. So, if you look at some old commits, you might get a situation that it seems like none files are highlighted because it is not in the repository. Yeah. So, it's difficult to spot, but maybe it's easier to do it here and I can basically make it a visualisation a little smaller and make it a little wider. So, then you can see actually see more, right? Yeah, and now we can revert this process and go to the other filters. So, you have also this author philtre. So, if you have a fix and then you have the list of all authors in this repository, so you might be just interested in what this person did. If the text filter is not enough, then OK. My computer is not very fast, but then I will select another person and those two people who would be only included in the list. Yeah. So, you maybe I'm interested what those did because I didn't find my commit, so I can exclude these people. I know that they are not authors of the commit I'm looking for. And here you have the date. So, the first role of the date is informative. So, if I change some of the filters or do whatever. those dates will change. So, they actually tell you what is the oldest and the latest commit that you found. So here, now we have this date. And if I change it, we get another date. It is also because of this filtering through all the commits, so you want to have some kind of boundaries. You can also set it to any value. So, let's say that you are interested only what happened in this year. So, you start from here and then this list has changed and we will never update the state without you before you reset

this view so basically closing and reopening that. Then, we'll come back to the default behaviour. Yeah, so that's more less the presentation that I wanted to give it to you. And my first question is connected to this UI for the commits. So, if you think about the UI, I showed you. Do you agree with the statement that the commit messages, so the information about commit messages is presented in the clear and easy to navigate UI?

Denis:
Interesting, but that's about this commit history? If it's in the clear and understandable?

Dawid Wozniak:
So if I went through all this stuff and then explained it to you how to get it done and work and what you can do here. You want to play with it, just go. So like sorting in the first look then you can think if you like it you r basically not it is that kind of question.

Denis:
To me, it's a little bit crowded like I'm not sure if you can you… OK it's by date, there is an example like have multiple.

Dawid Wozniak:
Yeah. So, then you need to remove some filters. Users still applied. Yeah. So, if you apply something, then it persists.

Denis:
Ohh. I would say like those long messages like the very long ones it's…. I think they're making the view very inconsistent like for my eye. Like you can see like in on the 2nd row here.

Dawid Wozniak:
Yeah. So you mean that we are going to the next line?

Denis:
Yeah. Like to me, these dates, like if I just look at it. Yeah, I can see the errors, but I cannot easily match error with the date. Kinda. You know those groups… I cannot see exactly if it's a group one, next group or not, just from like a glance basically. Hmm, what else I would say? Yeah. The fact that, like, some groups are way longer, like the messages on the on the next line, it's bad. I mean, yeah, from implementation said I understand why, but still.

Dawid Wozniak:
Now you're talking as the user. We wanna see your thoughts a user.

Denis:
So, I would say look maybe a bit more space between the groups is needed because the group by, it is maybe a bit more space between them would make sense or actually another thing is that, well, since we are here you can actually see what happens if you click here. OK, you go to the commit. Yeah. I would say some feedback like visual feedback on whenever you navigate here on this menu would be definitely nice. In general, because just the showing the core list that changes, it's not always understandable if you can click it or not like you may think. It's a mistake because nothing else is highlighted,

Dawid Wozniak:
Yeah, like they are very nice suggestions and if you needed to give a grade like from zero that is completely not usable to you to ten you that it's like perfect UI. Nothing to change. What grade would you give to this UI?

Denis:
It has a lot of filters like when they show me sorting, I was like hmm, can I search for things? And you're like yeah, very it is there. So, I think like I would give it like… seven-eight, yeah, seven, maybe seven of ten. Like it has nice functionality but like from this feedback perspective as a user. More introductions would be like better and could be faster. I would say also I really don't like I mean the fact that like it loads so slowly, and it is just because of this animation or like visualisation there. Everything is in memory and the mouse moves so slowly like I believe that's another problem that you've seen to understand.

Dawid Wozniak:
It is not topic of my thesis and there are some ways to solve that. They are just really time consuming. Yeah, but it's a good transition to my next question. When you look at this product, it is what we have now on the screen for some repository that you might know do you think that it would be beneficial to use this tool for any of your work project or side work project, maybe. something for academia and then you can also think in the terms of the grade so on the scale from zero that you will never use it except for this presentation to ten that you can use it for all your work and side project multiple times a day. What would the grade be?

Denis:
Only the commit part?

Dawid Wozniak:
No, now we are talking about the whole tool. You can also say something what you mentioned that it is slow during the previous question, you can give some additional comments about that.

Denis:
First, I would say I really…. like in my previous workplace. I really like this tool GitLens. I think you can have like… you're in a file and you can see exactly who is responsible for each line. I think I would use this for something similar to that, but like on a bigger scale like, at least, one that can look at multiple files on who's responsible like that's very good for overview. Look who's responsible for what file. So, I can go to a specific folder and like okay who made this component and like who was doing this right or what kind of files were changed. I don't have to go to every single file exactly to see who made more changes, but to the folder what I really like about this one. What else I would say? I think it would be really nice like as I can see that you URL here. Now, it is only for this repository and the name of the branch. When you go to a different folder, I think that it would be really nice. If you could also modify the link that like on the browser…

Dawid Wozniak:
for now, we have this navigation here…

Denis:
Yeah, I can see that reason for purpose.

Dawid Wozniak:
The idea was that for smaller repository. It's actually faster when we don't reload everything with the UI.

Denis:
It's so, I don't know again from implementation technical perspective what you use.

Dawid Wozniak:
Yeah, it's not important here.

Denis:
Just wanted to say that it was done in the way of reloading it every single time. I understand this perfectly. My point is that I can see. When I have a huge amount of files and components, I'm like, OK, I have this 10 components, who did them. I don't wanna go to a file and I don't wanna go through this UI every single time when I have to go to this folder and then to this folder and then and then like 10 layers, levels, like that. I don't really wanna do it. I would rather say like okay open this folder in this view right away. Yeah. And that's why I mention the navigation. Like if navigation would also update the kind of URL, at least I was doing Angular and in Angular, you have a nice service which you can use for updating the UI completely and you just update the URL. So even without the reloading UI, the tool just updates URL and it will save it in the history.

Dawid Wozniak:
Yeah, well, that's nice. So, if you need to give it to a grade, if you gonna use it for anything at all, what would you say?

Denis:
In my previous job, I'll probably use it like at least third times more. It's very hard to say. I think my my grade would be like 2 or no, it’s too low. Like three out of ten. I would definitely not use it constantly to see who is responsible for what, but when I'm even here when I'm trying to develop something new with some new feature with classes or like other things I haven't touched before. I would first like drill it down something like this tool saying, ok, we have this module - names of whoever I see then we have these features and I'll see what is developed under it, who developed that. And yeah, that would be definitely easier than the current way.

Dawid Wozniak:
Yeah. So, thank you for that. You gave it three, good grade. So, it means actually that you will use it for something. Yeah. And when you use it for something, then I will come back to this commits view. If you think about it, what would be your primary usage of this stuff? So, in other words, what would be the reason that you will go to this tab, what scenario you can think of to fulfil with this view?

Denis:
It's a bit hard to say what I want do looking for maybe. Like basically, my first idea, what I would do is that I have a file and I first want to see - What? Where? Who? You have already showns so like I have a file and I want to see the recent changes to it. Sometimes I remember even at my last job I had this issue. I have a bug and I have to go a few commits back to see what has been changed like exact lines basically. I don't know if it's possible. It would be awesome if I would go into commits and be like, OK what change, maybe you can show what changes are there is like a thing in the summary, you can get that like - how many deletions? How many insertions? So, you can see like how much contribution they bring like how many changes there was in total in the file. I think that would be nice.

Dawid Wozniak:
About your usage, so your usage as I understood it is to go here and look for some history of file to find the commit that introduced some regression.

Denis:
Because, well, I remember it was such a nightmare. I had to go to like 10 colleges before, like, back to to them see how the line was changed to understand like, why the bug exists. And then, like, I just reverted the commit that fixes the bug. I don't know. Can you like when you click? It's very like when you click to a commit, you go to right details. Can you open like a comparison diff?

Dawid Wozniak:
No. So, it is available externally it was one of the idea but we decided not to. Don't do that as it would require some extra tool, I mean here, people like to not overcrowd things. We have talked about it, but we didn't find that useful during previous iterations, so there was some iteration with testers and they didn't mention that these will be useful, but maybe it would be useful for you. So, I think that you already started answering my next question. So, it is like what is still missing regarding commit history? You mentioned the commit size in terms of the lines. Is there anything else that you think that it would be nice to have in this view, some other sorting, filtering, other view?

Denis:
Let me think about it. We can find a file. Is there anyway that I can navigate to the file without using this visualisation?

Dawid Wozniak:
Yes, so you can do it here. OK, so there is an option to do that. It's also slow. If I started now, it would take a lot of time, because we basically brute force it. So, we go through all these files and then released a full list of all that match some part of the files. So, it's not perfect, but yeah, it's this issue that I mentioned at the beginning. For the huge projects, it's not that perfect. Now we are still working on it. In small project, it's more useful.

Denis:
If we are sorting by date, we show people, right? When we group by day, maybe actually would be nice if you group show an author who made it. To reduce this number of clicks that required like, OK yeah there is a please merge and I click here, and I can see who created it kind. So, I'm more information with summary.

Dawid Wozniak:
So, we have this metric here top contribution. Yeah. And then we want to reuse those colours. The problem is that it becomes very colourful so here we have like 93 different colours and it's difficult to distinguish. Here, the solution was to do author sorting here. So, then you can actually see OK all those commits belong to this person, maybe it's not perfect and maybe we can improve.

Denis:
But when you sort by author, does it also sort by date?

Dawid Wozniak:
Yeah so, it is the latest commit here.

Denis:
I like this way. Because when you have a hierarchal order, that's fine.

Dawid Wozniak:
So, it was my last predefined question about the process, but maybe you have some general comments that you would like to say. Something that I haven't asked about and you think that it's nice to actually comment on something that is maybe not connected to the commit history. Maybe, it's just the product or your general thoughts. It's optional. So do you have any?

Denis:
Basically, I want to kind of have a discussion with this one because so these two are primarily for as I understand like visualising the repository?

Dawid Wozniak:
correct

Denis:
And you add the history to it. Yeah, we can see our compute contributors. So, we can't see it by ourselves. Which files, which person did basically what. That's actually nice. That's very nice, I think.

Dawid Wozniak:
It's OK if you don't have more.

Denis:
I do not have on top of my mind. Yeah, that is this like, yeah, the only the only thing I would be concerns about like this performance or this ability to just go to directly to a folder.

Dawid Wozniak:
Yeah, that's the scalability. So, I'll be referring to that in the future work for this product in my master thesis. And now I have a few questions about you. So that I check how your answers correlate with people with similar career factors. So, the first question is how many years you have worked professionally with IT projects, including part time jobs and it is in years so you can need to be like precise to months.

Denis:
Yeah so, it's five years.

Dawid Wozniak:
Yeah, that's good. And what is your primary role?

Denis:
Wait. It's not the five years. It's two and half there and here it's almost a year. So, it's like three and half, not five years.

Dawid Wozniak:
Ok, and your primary role is….

Denis:
software developer, software engineer

Dawid Wozniak:
OK and if you think about your average team, how big it is, including like developers, testers, so QAs, product managers, product owners, but not the high management like marketing people or manager that it's doing some other stuff.

Denis:
From 6 to like 10 people like up to 10 people out. So, 8 to summarise it like that.

Dawid Wozniak:
And if you think about your average project, do you know how many commits are there and you can say per day, per month, per week and I mean on master. So, if you have like merge and squash strategy, it means one commit per pull request, but you can maybe have different strategy that you have more commits and it is also rough estimation.

Denis:
I cannot say because… I can say how it was in last in my previous one. Ohh, we kept master for just deployment stuff, so whenever we push the master, we will start the deployment. So testing,

deployment and all the hotfixes, hot fixes will also be pushed to master right way without any pull requests. So, we have quite a lot of commits on the master branch we had quite a lot per month. I would say we release at least four times a month and then that will be like every release we have, I don't know up to 5 commits, let's say.

Dawid Wozniak:
And see if you answer the same question in terms of the files. So here we have like 36,986 files. So, if you think on average how many files are in your project, what would you say? It's also like the estimate.

Denis:
Yeah. OK. I cannot estimate that here. I mean with my current project what I'm doing here because well, it's CI we don't know how many files are there.

Dawid Wozniak:
Yeah, it's also a good answer because I know that you don't know that by heart, and it's also the answer, but also classified you somehow.

Denis:
There is another thing. For instance, if you go to our repository or like the way you create this great code, you can create like a project, but this project is not included in in the source code. It's included in like a module as I understood and then like multiple people could have created like their own projects locally, but in the end it goes under the same module.

Dawid Wozniak:
It is like this concept of the one big repository like mono repo.

Denis:
Yeah, I guess I'm not very familiar with a lot of terminology. Even if like what I'm doing here, I'm doing right now is that we have, I don't know, let's say not that many files like in my module, there are like 20-30. All the files I would say, but then like we're more modules you are working with, like with tests and then with another one. It increases and increases and increases basically and then that's what we are not touching, definitely not touching so many files. So that is this. So then my last project, just very fast, my last previous work I would say so I was working with Angular and with Angular we were using some principal where you have like if you're writing a component so like let's say. Some model dialogue, right? You have first a component then wrapper component, component which loads some data or uses services for calls. The inside component is like a dumb component where it just gets the input. It has some inputs for the data and some outputs. Maybe for some events, but it doesn't know anything about all these calls and whatever and it kind of simplified architecture. So, you have like a smart component and dump component. So, you can imagine for every single like button or not bottom. But like for every single dialogue that we had for every single pop up, we had at least two components and of course dialogue could have like multiple views, multiple buttons and so on that looked like really nested. It is quite common. It is quite a lot. But it's also like the structure conversion. Yeah, and then we had like at least four files per component so. even for something simple, we had at least like 8 files. I think it was like the size was like around 2 gigabytes, which is like compared to this not really that much. Ohh but I think what that was definitely more than like 1000 thousand files.

Dawid Wozniak:
OK, so you that was my last question, I think so now I will write my master thesis. I have some chocalte for you. So, you can choose which one you want. If you don’t like any, we just find anything

that would be nice for you. Then we can arrange later. So, thank you for participation. And my last maybe another question, but this information is that when I publish the paper, so my master thesis, would you like to read it? you can say it now. So then I will also send a copy to you. It is being an article format. So, it should not be that fifficult to read. OK, so would you like to have that in your inbox at some point?

Denis:
Yes, sure

Dawid Wozniak:
Thank you. One more time for helping me and I will stop the transcription right now.