Dawid Wozniak:
I started the transcription right now and you can see on your screen what we are saying. So, let's start with the presentation of this tool. So here we have the GitTruck and when you run this tool, you can see there are some names of the repository that you have at the current location. If you pre-analyse them, they have this label, otherwise you need to analyse them. It takes some time. So that is why preanalyzed some of those and we can see Git track with Git track how the project is done. Now it's basically reading the file that was created and then we have on the left side some information like the title of the repository, branch that is analysed. We can change the branch. We can see when it was analysed. The short version of the commit hash, how many files there are. It's some feedback box that we are not going to talk about it because it's not important and we have this visualisation. So. it is a bubble chart. The size of the bubble is meaningful in the sense that if it's the biggest bubble in the repository, it is also the real size of this file. So. it means that this file is the biggest in the whole repo. Do you have any question?

Benoît:
Yeah, indeed. Do you prefer that I asked question during the interview or after?

Dawid Wozniak:
It depends how you would like to do that. If you want to ask any question now, I can answer. Otherwise. you can go through the presentation, and you can answer your questions then.

Benoît:
Thank you. Maybe I prefer just go directly to question, so if this visualisation shows a file, then size of the circle depends on the size of the disc storage, right?

Dawid Wozniak:
Yes

Benoît:
Okay perfect.

Dawid Wozniak:
Yeah. How do we define the minimum size? So, the minimum size is that the smallest file in the repository. So here, it's probably this file and then we take some ratio. So, if you compare this bubble with this bubble, you can see how many times this bubble is bigger than this. It should reflect the real ratio between sizes. We also have another view that is called tree map, but the same rules apply. Here we just have the squares. So, I think that the bubble is nicer so as you can see, there are a few colours, and these colours are colour coding something. So here, we have the file extension. In this repository, we have those file extensions, and this colour means TS files. I can change that so we can also have number of commits. Then you can see which file was frequently changed, let's say, here „analyse server" something - this file was changed only 10 times so it's not that bright as another. When it's brighter than others and then we have less changes. So, another metric is last changed and here if it's lighter, it means that there was change sooner than later. This is actually defined here. That was like probably never changed. We have a single author. So here we have two groups. The red groups says that OK, this file was committed just by the one person and other files, they have multiple authors. We have top contributors. So based on the number of commits, we say that this file should be owned by this person. So here we have Dawid, and we might remember from the previous metric, that it is the single author of this commit. Here, we have like Jonas, but we have a few authors, so it's kind of enriching single author. I think the track factor, it is how many people contribute to this file at all. Also. it's kind of represents the same information in the different way as

here we have one author. Here, we have five authors. So, it just depends on what you want to see. So, we just stay with the file extension and then we have chart depth. So now, we have a full chart depth, but if you set it to one, it is what you would see. It is like you go to this folder in your File Explorer. Then if you click there, we just follow your choice.

Benoît:
Can you just show me again this last feature?

Dawid Wozniak:
Yeah. So, it is a chart depth. At the beginning, it was set to full. So, you can see all the files in the repository. Then if you set it to one, you can see only the first level. So, what you would see if you went to the File Explorer, then you can increase it to two to three and so on. So, you can see how complex your project is and where the real files are located. So, you might have very complicated structure and a lot of folders. Here, it is quite small project. I will stay with full. It's not too crowded. Finally, we have the detail view. So, if I click on the bubble, let's say this bubble, then I have some general information as it is a folder, it says how many files and other folders are there, where it's located, authors distribution. So based on the number of commits, we calculate who contributed the most this file and list in the percent and the short commit history view. So here it is not clickable, it's always sorted by dates, and you can have glimpse on it. Then, we have the commit part, and it is the part that I am responsible for. So here we have some more option to filter and sort them. So, the first thing that you can do, you can change it from latest to oldest. You can make it from oldest to latest. So, you can see what a long time was ago and then you can also sort by author. So, you can say, OK all these commits were done by Dawid, but I'm not interested in that. I wanted to see Jonas's commits or this Thomas' commits. So, it really depends on what you want to see. So let's stay with the dates and then we have some filters. So maybe you know some part of the commit message, let's say that you are searching for some „fix," so then you put „fix" here and you get those all commits that include „fix" in the message. So here we have some „fix". Maybe, it's not enough, then you can tick it. OK, maybe it's actually in the commit description and then we have more commits here, especially merge commits when we put some information what was there. Finally, you have two other major filters, so you have authors. So, you can say OK, I think that this commit was done by Dawid and then you have only commits, that were committed by me or you can exclude Dawid. So, I'm pretty sure that it was not Dawid. So, I can exclude this person and you have the dates. So, if you don't change anything, they are kind of meaningful. So, they basically tell you what is the latest commit that you found and what is the oldest. So, if I change some thing here like I and think, OK, that's doesn't change anything, but if I remove this filter, then you can see that it is from this day to this date and then I include Dawid and it changes. So. it has informative role at first and then you can set it to something and then we will stick to that. So, we will not change it anymore, so you might just think. OK, I'm interested from the first April. Then even, if you change something there, we still keep it as you set it. Finally, we have a small checkbox. So, you can say, OK, I'm actually not interested in this merge commit because they probably repeat some other commits that are there. So, if I uncheck that, they are gone. Yeah. So then let's say, that I was interested in this commit, and I found it. I can click it and I can say okay this is the message, the description, actually there is none. There is a hash, if I need it for some cherry picking or whatever, when it was exactly created, by whom and I can „show edited files". So here, we check which file we changed with this particular commit and we highlighted them in the visualisation so you can see them here. You can also see the list file what was changed with this commit, if they are still present in the repository. If there is a very old commit, you might get a situation that this list would be empty because we reorganised everything. So that's everything. I'd like to present my first question to you if you think about this. Do you think that the information about the commit messages is presented in the clear and easy to navigate UI?

Benoît:
About right part?

Dawid Wozniak:
Yeah, yeah. So about commit information. So, the one commit view is here just in the general tab and then you can go to this „commits" tab that is more advanced view for commit history.

Benoît:
I think it's really useful based on just your presentation and I think that we can easily find some specific information more visual that "git blame" or git something to show the history or something like that. So, I think that this could be really interesting. To search specific information easier than in other tools.

Dawid Wozniak:
OK. That's good feedback. And now when you say that I would like to give it some kind of the grades for the UI part. So, from zero that it is like not usable, you will change everything to ten that is like perfect UI, nothing to correct, don't change anything, it's perfect.

Benoît:
Is really responsive and reactive? I will just want to use to keyword. It is more reactive based on to demonstration. I think that some stuff could be improved from UI perspective. For example, „to" „from" those two, maybe you can just use a slider probably. It's more visible, more dedicated to UI experience from my perspective of course. So, I think I can put eight, but including my remarks.

Dawid Wozniak:
Yeah, that's very nice feedback. Thank you for that. And now I presented the whole tool to you and hopefully you get a little of a glimpse how you can use it. So, if you have access to this tool, how beneficial it would be to use it in your work for any purpose or side project and also, it is a scale question from zero that you will never use it outside of this presentation to ten that you would use it for all your products multiple times a day.

Benoît:
From a perspective where I'm using a project alone, I'm not sure it's this kind of tool would be really interesting. I think that such kind of tool is really interesting when we are many collaborators on which you can see here in which part each collaborator works. I think it would be more efficient and I think that's another use case where I can see really interesting your work when we need to evaluate to work of students for a project. So, for example, I was invited lecturer for specific course where I teach to develop a website, okay? My students who works in groups, need to create a website, we must have to analyse different contributions of each project. I think that such kind of tool is really efficient to see if they contribute almost similar or not balance the workload. I think that's such kind of tool is very interesting for this specific use case.

Dawid Wozniak:
OK, I think that is a perfect use cause for this tool. So, if you needed a great from zero to ten, what would it be?

Benoît:
In general or for efficient use case to evaluate students?

Dawid Wozniak:
It's for you so you can decide by yourself what is more appropriate.

Benoît:
I think I will put a five. If I consider the different repository, OK, but I only put good. For the specific use case of assessing student, I think that is really useful and I use it there.

Dawid Wozniak:
OK. So, thank you for this feedback. And now I have another question. So, you mentioned that it would be very useful to use this tool for grading students for their project. And so, if you have such a scenario, so it was a very clear and nice scenario. The main purpose to go here to commit, would it be to find the distribution of the authors? It is at least what I understand. So, is it right that it is the primary usage of this commit view, or you have some other primary usage for that that can be fulfilled going to sorting and filtering of the commit history?

Benoît:
Um, can you just repeat the question? Just to be sure.

Dawid Wozniak:
Yeah. So, the final question is - what would be your primary usage of this commits view? This what we have on the screen.

Benoît:
If I didn't answer the question, don't hesitate to say, OK, it's out of scope. OK? Just, you can explain me again the question I think that's needed. Can you just explain again to question?

Dawid Wozniak:
Yeah. So, if you use this tool in the most common scenario, what would be the reason that you go to this commit? What kind of information you would like to have from it in the current stage? So maybe, let's say, I wanted to see a student project. And did Dawid do anything in this project? Actually, okay he did. So, it can be one scenario or there might be some others that I can imagine.

Benoît:
Maybe I have a remark about sorting… It's in general, I think. You just propose an author distribution but your authors depend on the number of commits. The number of commits doesn't really reflect the real work.

Dawid Wozniak:
Yeah, that's a risk.

Benoît:
Because a commit could have only one author, commit can be changing all architecture. It depends on who are the students in special, specific use case. So maybe, I think that could be interesting to put a disclaimer on this specific part of autor distribution. Because, of course, you cannot manage that correctly if the end user doesn’t use Git correctly.

Dawid Wozniak:
Yep, that's true.

Benoît:
About messages, I think that's really visual. I think that's really interesting to see. How are we gonna say that to pair programming? So, for example, assume that students work in pair programming. Okay, so I'm working with someone, and they put that in GIT with whom they worked together. Do you plan to consider this into authors?

Dawid Wozniak:
As far as I remember, the way how we did it, I just need to find the right commit. Actually, there is none, but if you have this author on GitHub, we also calculate them and we have like plus one to this person and plus one to another. So, when you have author on GitHub, it is okay but some people don't know how to set it up and other stuff, so then, it's not 100% correct and it also should be taken into account for an author of distribution because it is based on the commits. So, something that we have from users.

Benoît:
Yeah, of course. Your tool is based on the on the commits, so well, of course, again if you don't use GIT correctly, you cannot use probably your tool, but finally it's the goal of using your tool correctly and using Git correctly. So that's right. Yeah, I think that's it's okay for me if you have this kind of information.

Dawid Wozniak:
OK. And now I have another question. So, we play a little with that. So, do you have any ideas what may be missing regarding this commit message history view? Something that you think should be here, but it's actually not there. So, you mentioned a few things that might be interesting to have, but if you have something more, that's opportunity to say it.

Benoît:
Not specifically on this part. Maybe, I have a comment on the visualisation itself.

Dawid Wozniak:
Yes, yes, please. So it's good feedback.

Benoît:
Can you show me again the metric change, for example? Can you show me again? Yeah, indeed. Perfect. I can see here the number of commits depends on the colour, if it is darker. It increases your number of commits. OK, but if you change your metric again, and I think that you have a track factor or something like that, I think.

Dawid Wozniak:
Yeah. Then you have the reverse. So, the number one is darker and I can explain why. So, we set it that if you have something that it's light, so like this, then it means that it is good practise. And if you have something, that is darker that is a bad practise.

Benoît:
When I see that was not intuitive, I say, OK, it's maybe a counter intuitive but OK, based on your explanation, I understand why you use a reverse colour. Yeah, I think that's really interesting tool and a specific use case that could be interesting is the assessment of student project.

Dawid Wozniak:
Okay, that's great. So, it was my last predefined question, but if you would like to say something about this. That should be included in our meeting, that's time for you to put some comment or some suggestions for the general tool and if you have anything like this, please say it now. Otherwise, I have a few questions about you. So, I can see that maybe based on some factor, I can correlate you with other people with the same answers.

Benoît:
I think that could be really interesting tool. For a moment, I don't have specific comment on improvements based on what you show me, except maybe a change to the dates.

Dawid Wozniak:
Yeah, I remember that.

Benoît:
That's really a good work.

Dawid Wozniak:
Thank you for that. And now I have a few questions about you. So how many years have you worked professionally with IT projects, including part time jobs and it is not question that should be answered very precisely, you can answer in years.

Benoît:
I will answer your question based on when I really started to work. I didn't include my study. It's a long time ago before, but when I started to work… it's 7 years.

Dawid Wozniak:
What is your primary role?

Benoît:
Primary role is a researcher.

Dawid Wozniak:
OK, that's great. And if you think about your average team, how big it is including developers, testers, product managers, product owners, it can be open-source product. It can be like student projects or on the project that you leaded as a teaching assistant or something like that. So, you can basically combine all of them and think, what is the average team size?

Benoît:
I think that's I manage project who I am alone till project where I'm project leader as teaching assistant. I was project leader. I have students' project. Eight students per group. And when we need to merge together, I think that I have 24 students.

Dawid Wozniak:
OK. And if you think about the average product that you do, so you created something or you participate to some projects, do you know how many comments are there and you can say per day, per week, per year, per month?

Benoît:
Ohh I cannot answer that. It is totally different so I cannot.

Dawid Wozniak:
OK. That's also good answer. And another question if you have your average project, do you know how many files there are there for average? So here we have like 112 files, or do you know?

Benoît:
This depends again on the project so I cannot say that because I proceed to a lot of projects from alone projects to assistant for students. I'm manage my thesis project that was really big and it depends on, so I think that's the question maybe for an industrial people. It could be easier to answer and but in my case. It depends on your skills and if you like to refactor. Personally, I prefer to have small files with a really nice refactoring. That here has a lot of big filse. I cannot answer. I don't have a specific knowledge.

Dawid Wozniak:
Yeah. No, no problem. Don't know. It's also a good answer. So, it was my last question and now I would like to just say that at the end, based on the evaluation with users and some research, I will publish my master thesis on the topic connected with the role of commit messages in the visualisation tool. So, if you would like to be informed about that, I can send it to your e-mail. It would be in the article format. So, it's supposed to be not that long and maybe it would be helpful or just interesting for you to read if you'd like to have that, just let me know now. Then, I will send it to you then.

Benoît:
Yeah, but it could be interesting to read your work, when you publish a paper on this work indeed.

Dawid Wozniak:
OK. Thank you for your participation.