Dawid Wozniak:
Yeah, so I'm going to present the changes that I created in this iteration and then I will ask you a few questions. So, I think that it should go smooth. Yeah, it is a third iteration, you can see my screen, right? So, in this iteration, there are a few changes related to the UI and the functionality. So, the one of them that you can see on the left side, it is that you can define the chart depth. The problem with that, so a little problem, is that you still have the bubble size meaningful. So, even if I set it to one level, so I supposed to just see the files that are in the current location. So, in the root of this product then you have a lot of empty bubbles with the size corresponding to them. Hopefully it will open. So, it is the one thing. Maybe when we wait, I will talk about another feature. So, another feature is that you can resize it so you can say, OK, I would like to make this visualisation smaller and now it's regenerating it and we can make it wider. So now, it's possible you don't need to do any magic, you can just do it by yourself using this points. It's now there. So, let's say, that I'd like to just have the one level and you can see, that now all these files, that were in the bubbles, were inside this bubble, are gone. You can see that there are a few files, but those files are very small on the root level. You have some folders. So, if I go there up, you have a very similar situation. So, I'm just thinking how to solve that to have still meaningful information about the size. So now, you can see that the „house" is bigger than „modules". If you go to „apps", you have more and more empty bubbles. It doesn't look that great. If I'm back now to full mode, you can see that there are actually many, many files inside. Yeah. So, it's something that we are still working on, but let's go to this commit detail. So, I have those commits and now I have sorting. So, you have a sorting by date in the accessing or the descending order. It is just the default sorting feature, and you can sort by author so you don't need to go manually and say OK, I would like to see this person and then another, another. You can just go there and see, OK, this person did that, and other people did this. Then, you can basically collapse them, and you can see all the changes. If you go to filters, now filtering here is not case sensitive. So even, if I put „ADD" like this, you can see that it's also detected this. So here is upper case. Here, it is lower case, but we still have there authors, so this is the list of all authors in this repository. We don't specify just to this particular level that we are checking, but to the whole repository and you can check some people. So, let's say, this person did, did they something there? OK, not really with the „add" filter, maybe without this filter, let's try to remove that. Yeah, there are some commits. That's good. But maybe I would like to see, all people, but this person's commits so then you have this exclude option and for the dates I added also small updates. So now you can also specify the time and here the dates have two meanings, one is informative. So now when I didn't set up anything and I put some extra filters so I would like to have something like here. You can see that this date changed so it shows you that the first commit was committed that date and the oldest is at that date. If I remove some of the filters, then this range changes. I can also set it to something and then it's fixed. So, if I set it to thi,s we detect that you actually done that and then when I even add some more filters. It does not change from this particular view. The last change that it was requested. So here, when you have the commits, this is one single commit we can see. There is the title of it. It is what was put in the description. So, the description of the pull request and then there is hash of this commit, when it was exactly created, by whom, and you can click to show edited files. So, if I did that then we are fetching which files were changed with this particular commit and we will send it in asynchronous way to this visualisation. Then, everything else should be greyed out. Just the files that were changed and I'm not sure that it will be that much visible because the size of the bubbles are very small. Yeah, so it might happen. But if we go, somewhere, lhere when the bubbles are bigger and I removed this and include all, then let's try to visualise this. It should show how it would look when there are some files that were edited. Now it's doing reverse so I will try to do this and in the meantime, I will ask you what do you think about this new UI stuff?

Ayrton:
Yeah, I think it's good. I think that maybe you need to put some focus on like how those options are presented, right? I think the options that you give are really useful. So you know being able to pick a date range and filter based on like include or exclude these users and then have this view on the side sort of matching up to what is being shown on the left hand side and the fact that like you have taken this UI and moved it so it sits on top of everything and you can change the size of it. All makes it a lot easier to use, but there's some stuff like the filters, for example, like the users are not sorted in order of the name, so if you try to find somebody in this list of users, it would be kind of difficult to find someone, because it's almost a random order and things like picking the date range. It's like two boxes where I need to go into the calendar and choose the dates. But if it's just a slider, that I couldn't move around, you know, simple usability things like that right? Like make sure that the drop downs are in a sorted order, consider different types of UI like a slider instead of an input field. All of these buttons and filters that you've got. They take up a lot of space, like if you are looking at this UI, right now, you can see that like I don't know maybe like 90% of this dialogue that you have shown in this filter options. There's a very small part of it that is dedicated to the actual results and the bottom.

Dawid Wozniak:
Yeah, I think that it depends also on the results when we go level up and then it will be more, I hope more commits. Then, you have the reverse situation. So, you have a lot of commits that this is a long list that you can browser.

Ayrton:
Yeah, but essentially you have to scroll all the way down and then the filter state is at the top right. But maybe like you could put the filters in some sort of collapsible header that always stays at the top, so you're like right now you have the filters showing all the time and then the whole dialogue with all the results is scrollable and if you scroll away from the filters, you know, scrolling down through the results and you scroll away from the filters and in order to change the filters, you to scroll all the way back up. But you know, if this was just like its own collapsible box like the filter stuff. If you could just press a button to hide all of that and only show the results, then keep them pinned to the top. Then, it might be easier to use in that way, right? It's like a lot of very small UI tweaks that you could make to take the functionality that you've built and make it much more usable.

Dawid Wozniak:
Yeah, that's a good suggestion. And so now. I would like to repeat the question from the first iteration. So. when you have more filtering and sorting, do you think that browsering is enough or you would like to have some more insights from the commit messages for now. So just to focus more on this commit messages and maybe include some parts of the description.

Ayrton:
So if you click this „show edited files", what happens? Does it just highlight them on the left-hand side, or does it also show you the files on the right-hand side?

Dawid Wozniak:
And so for now it just highlighted on the left side, but it was said to me that it would be good to have like a list here what was changed because now it's not visible. You need to go to the different folders to check, actually, if this file was changed or not., so it's better to have this list here as well.

Ayrton:
Yeah, because you could put something like, if you click „show edited files", it highlights on the left hand side, but it could also show you know like when you do *„git diff"* short stack and it gives you like the list of all the files with how many lines were changed. So, you get that but then you group them by directory then you could click it. If you could list all the files on the right-hand side that changed just part of this commit view and then each of those files and directories you could click those and it could zoom you to the view right? So. if you click the folder name, then it could take you to the view of just that folder, because maybe I wanna drill down through these changes to see what exactly was changed and a lot of the time, let's say, that I'm viewing all the files. I want the highest level view like show me the top level folder that contains all these changes. Then, if I made a change in the top level to the configuration file with every other change that I made. it was super small file down the tree, then if I say, show me edited files with these changes, it's probably just gonna show me every time this file in the repo, right? But a lot of those files are folders that live at the root level but change all the time as part of every pull request where, you know, If I was looking at some type changes, I don't need to see the changes to *package.json*, so we understand the other changes that were made, right? Cause that *package.json* is in the root and it would be kind of zoom out too far. So, if I could drill down to, you know, if I saw that I had one file, that *package.json* and the rest of the changes were all in some deep dwon folder. If I could just click that folder on the right-hand side and it would zoom the circle diagram view to that folder. That would be nice, right? Cause, it's like a list of files on the right hand side. It will be a quick way to navigate around on the left hand side without me having to click through this diagram.

Dawid Wozniak:
Yeah, and that's a good suggestion. But coming back to the commit messages and the description, do you think that there will be something more useful to take out from it than just browsering?

Ayrton:
No, I think that information that you have is enough, right? Because you have the author and the hash, which lets me find the commit easily. And you tell me when it's created and the message that went along with it. I don't really think of what other information, other than the files that we've changed that I would want to see. So. this is good. And I remember before, like you had this, but it was just the title like the very first line and the author, I think this view is much better. You know, I've been able to take that and drill down into it. So be that shows me more information. It's great. I think this is a big improvement comparing with the original version that you showed us.

Dawid Wozniak:
OK. And if you think about the sorting and filtering, you gave some suggestion connecting to the UI, but do you think that something is missing there so you can think about some crazy ideas like maybe sort commies in some other ways than normal sorting, maybe just by some property of the commit? Is it something that you would like to have?

Ayrton:
I mean, I think in terms of finding my way around, this is enough, right? It's like being able to see who changed what and when and I mean that you could add some stuff like show me the commits that changed the most files, for example, right? So, like you know, if I wanted to see what the most important changes are or what are the most sweeping changes to this folder, maybe that would be measured in terms of like the most files that were changed or the biggest change to the most commonly entered file or something like that. Right? So, some kind of that because you get a lot of commits to change this resource string. So, maybe somebody went into this extension and they changed the captions or something I don't really care about that…

Dawid Wozniak:
And so is it important for you to know the size of the commit that this commit is small and this is a big commit?

Ayrton:
I wouldn't want to suggest the metric that you would use because, you know, I think it would need to think about it a lot, right? But let's say that I was using this tool then probably, what I wanna know is either who is changing things in this repo or what changes often. If I change one thing what else changes or what are the most important changes that have happened in this area. Right? I'm looking at this extension and I want you to deliver to me the big things that have changed since the last time I looked at this. So, you know, like if somebody makes some big architectural change to how we render components or somebody made some changes to the Shopify extension where they, you know, removed some core functionality and upgraded to some other version or they introduced some new integration, it would be nice to see that. So, if I'm looking at this view now of every single commit that ever happened in this repo, there's gonna be, maybe four or five commits. that are really important for me to read and like two or three, sorry like twenty or thirty, that are just like small changes like „fix this tiny bug" or „change the caption of this field" to something else and those changes are like fluff and I don't really care about those, right? I don't need to know that this caption of this field changed, but I do need to know there's some major new integration was added, so if you could sort or filter this list by like biggest changes, you know. like changes in terms of most lines changed or most files changed or if you could identify which files were the most touched. aybe the biggest changes to those files? Things like that, right? Like you know, I don't really care so much about big changes to files that contain resource strings, because maybe somebody just went through them and cleaned it all up, fixed the typos, things like that. I do care about big changes to things like browser page session or big changes to things like the power automate controller and stuff like that, right? Because they're the files that have the core functionality inside. And if I was looking at it, I wanna know, like, who changed those, what else did they change? When? What are the biggest changes that happen to those like within some time period? Things like that. So, filtering based on impact.

Dawid Wozniak:
Yeah, so here there is metric this kind of answers your thing. So, you can see how many commits are there. I know that it's not perfect, but you can do some investigation which files were mostly done by commits. You don't know if it was a just moving around or maybe something that was bigger than that. I have another question so here when you have some folder, let's say, or some file. Let's say. this file, we have all these commits so we can also try to generate all files that were changed along with it. That includes just this file and some other files and then generate that. If you change this file, there is 90% correlation that you also change this file and something like that. I think that you mentioned that during the first feedback. So. do you think that is a good idea and would be helpful for you?

Ayrton:
Yeah, I think that'd be nice because what you see sometimes is that, um, like somebody writes, leaves a comment, like, for example, if we have CSS file that you define a width of something like some dialogue, you say this dialogue is this width and then somebody leaves a comment to say, if you change this line, don't forget to update this other file. But you know, like not all of those cases, are commented and people don't necessarily like make the corresponding changes because the comment is there, right? At least in this tool, it could reveal those links between files when this file changes 99% of the time along with other files, which tells me, as somebody who is working on this

file or working on this part of the code, like if I do change this, I also need to look at this other files. Even just to look at it, not necessarily to make a change, but like I should be aware that this file is like heavily linked and so if I make a change here. it's gonna impact this another one. If I make a change here, I need to make a corresponding change somewhere else that would be nice. I mean then you have some of these suggestions, right? I think it's important to ask yourself, like, are people suggesting things because you asked, what would you add? And they're just suggesting anything or are they suggesting things that are generally genuinely useful things, right now they're suggesting things that are within the domain of this tool. because it would be nice if, for example, DevOps did that right, if I sent a pull request and DevOps is like I can see that you changed this file. Did you forget to change those other files and they could leave a comment, or it could give me an e-mail notification or something like that. But then that is something that I would build into DevOps, and I would not build that into this tool.

Dawid Wozniak:
Yeah. So that's the idea that here we have developed, and we have the team that is doing that for you. But if it's something that is needed, then we might have some kind of extension for. let's say, GitHub or like DevOps stuff, then you can have this tool running somewhere in the background. And one option is that you go there, but it might be automatically running in the background and it updates with all pull requests this dependency list, then you will get some notification out-of-the-box without going there.

Ayrton:
Yeah. Yeah, I think that useful in general, but then that's the question, right? That would be something fantastic because I'm always finding myself in, you know, I send a pull request and then I forget to change one file. So now, we have a bug and then I send another pull request that's like change the files that I forgot to change in the previous pull request, you know, like I updated some width and some JavaScript somewhere and I forgot to update this file, so I have the main commit with my feature. And then. I have a second poll request that's like update the files that I forgot to update. But it if DevOps could want me in advance, if GitHub could warn you in advance, you should change these things. That is easier, isn't it?

Dawid Wozniak:
Yeah. So, I think that this is a good suggestion, and I would like to also confirm it with other people. So, then I will build it in this tool and if they all say it's some feature that we also need in the open-source project or in the small companies, then the next work item might be just to extract this feature to something smaller. Standing alone and doing just this one thing very good. So, then you don't need to think here. There is one thing that if you have the image because of the security reasons, you cannot display it in the description. So, you have some bad format links. I would like to ask you general about this visualisation idea that this is greyed out and only the files that were changed should be visible. So here when you change some very small file, one of these folders you cannot see it. You need to actually have this bubble big enough to see that it is not great out, so it is the one issue that I would like to also show you that it works like this.

Ayrton:
Yeah, I think that you would need to highlight those things until I was zoomed in enough to see, right. So, I think to be honest, like rendering the chat at the maximum depth is not always useful. It actually is better to have to drill down, right? Like the full chart depth is best because of things like this, right? It's like, yeah, I would expect in this case that you would not highlight all of the files just because I'm zoomed out, right? Like if I put full chart depth, you should only highlight the things that change. But I mean, you could highlight all of the folders that contain that file, right? So, you don't

need to hide, let's say. that I made changes in super deep folder, and I have this full depth set to one. You can highlight every folder in the tree that contains that file, and you don't need to highlight all the files that are consumed by those folders, just the one file and all of the parent folders, because then like at least. I can see which parts of that changed, right? So, I'm looking at this view and you selected a commit that only changed one very deep file, then only like the outside ring, this frame would be highlighted and then maybe this base bubble would be highlighted and then pages would be highlighted, but everything else would be gray out. So, I think it was actually this file. Like if you have changed the file, I would consider all of the parent folders to also have changed, not the contents of those folders, but the folders themselves. I would highlight those as they have changed, because I think that's clearer like in the view that you showed before. Everything was gray out and maybe there was something that was highlighted that was like in full colour. It wasn't right. You can't see it because you're not zoomed in. Yeah, exactly. So that's not good because I'm looking at this view and it looks to me like nothing has been changed by this commit because the thing that has changed is so small that it's not rendered on the screen anymore. So I think rendering the parents that are big enough to be seen and then, OK, I can see that like these rings are highlighted. So, I can't see the file, but there must be a file because they are highlighted so then I go there, right? Then as a user, I would zoom in to see the file that have changed.

Dawid Wozniak:
OK. And now I have the last question. So, you've answered this question already. Once when I ask you about this stuff, when we were doing the first iteration, but now when you have sorting, filters, we would like to show the changes, would you think that your main usage of the commit view would change to something more specific, maybe some other scenario, it just still browsering and searching for commits which regress something, it is mostly, what you said, I think.

Ayrton:
Yeah. So it's mostly for like seeing what changes have happened in this folder from a commit, right? So, the three main things is either I'm trying to debug something and I wanna see which files were changed by this commit, but I wanna see that. I don't want to look at the list of files. I wanna see it visually. The other one is like I have stuck on some bug. I'm working on some feature that I don't normally work on, and I want to know who to ask. So, I wanna see like, who changed the most files. Right? Then, the last thing is like I'm making a change and I wanna see like if I change this, what else should I change? So, I'm in some file and I know that if I change this, I need to change something else. So, I want to be able to go to this tool, find that file and then see a list of files, that were also changed and also see the commits that changed those, right? So, I can not only see that those files change, but I can drill into the commit and see how they would change, which lines would change, what change was made.

Dawid Wozniak:
OK. And so that was my last question, if you would like to say something about the product in general or something not related to commits. Now it's a chance to do that. If not, then we can finish this iteration wave?

Ayrton:
Yeah. All, I would say is, I think, you've improved it massively since the first time we saw it, right? So, there was a lot of information that I wanted that was missing and it was kind of hard to visualise that because everything was kind of on top of each other, right? And it was very cluttered. But I think now like you pulled this out to its own dialogue that you can resize, collapse and close it off. The actual visualisation can be resized so that, if something's covered up, you can just make it smaller and you've brought all the additional information like the commit hash, the description and things

like that. So even though, there's not necessarily like a link to DevOps, which I think is one of the things that I pointed out. Just like ID would be nice if I could click a link and it would take me to the GitHub or DevOps or whatever. You still give me the commit hash, so I can just take that and copy paste it, right? Things like that and I think l in terms of the improvements that you would make, the only things, that I would change is like minor visual improvements. They are usability things where you could take to the UI that you've got and condense it. You could make it easier to work with change to the slide or whatever but the changes that you made. They are great. This is like a huge improvement over the first version.

Dawid Wozniak:
Great, thank you for those words. So now, there would be one more iteration when I made some changes, we will make some changes to this product still, but then we will compare the first version without much information about the commits and the fourth version. We see some scenarios that it might have been more helpful without commits and might not have been more helpful. Then, I will probably bring some chocolate and some other things for you. Thank you for your participation.