Interviewer1

Yeah, the recording started. So I'll start with the demographic questions. So the first question is, how old are you?

Interviewee

37 years old.

Interviewer1

Okay, and how do you identify your gender identity?

Interviewee

Male, straight male

Interviewer1

And what is your highest level of formal education?

Interviewee

Master's in Computer Science?

Interviewer1

How many years of experience do you have in the software engineering field?

Interviewee

Down to Bali, 12 years. Okay.

Interviewer1

And how many years of experience do you have in leading teams?

Interviewee

Probably around three, four years. So these are all like, you know, like an architect role not, not like, people managing it's more. Yeah.

Interviewer1

So these are all our demographic questions. Now we'll jump to our interview question. So the first question is, what is your perception regarding a bug localization tool? Like when we use the word back localization, what do you think about that?

Interviewee

Sorry, can you explain

Interviewer2

a word? What is your perception? If when we say the phrase bog localization, what do you think it is?

Interviewee

Okay. I mean, two things, one, localization, meaning, bring it down to one small component. So it doesn't affect others around it. Sometimes, like we use the word localization, it ends up people think about languages. So those are only two things. Okay.

Interviewer1

So in this study, we defined a bug localization tool, a tool that can identify potentially buggy files, given the bug report or stack trace, or both. Suppose that repository contains 700 source code files, and based on the bug report, that to the shortlist, a few files, potentially responsible for the bug. That tools's output is a ranked list of the files sorted based on their probability of being responsible for a particular path, then the developers can start further investigation or fix produce fix procedures from the shortlisted files. That is our definition of a bug localization tool. Yeah, and given the state of the art of such a bug localization tool, would you use such a technique?

Interviewee

Depends on the project size, right. Like, I didn't last few years be in only startups. It's not that hard to identify the source of the bug. I mean, like, it will take time, because like sometimes, like if the bug is in like the data, you get the law to literally go through different data and see why this one is different order, right, like, but most of them fits like a code thing. Like, it's probably way faster to identify. I can see in big project, right, like I worked on "complex system". J generally, like sometimes a buck shows up, like, there'll be just teams fighting, saying you will fix it, you know, like, all those things will happen. It's like not ours, ours works like we wanted to. So it's your problem, like, those things will happen. So, yeah.

Interviewer1

So what do you think will be the benefit of such a bug localization tool?

Interviewee

And if it can get things, like, reduce the time to start debugging? That's obviously like a big game. Again, like it depends so much on the developer, right? Like some people, they work on multiple different things. And let's say they're working on something else, now they have to work, they get a bug on something else. It's a that's a transition. Like, as a developer, you you're switching. Okay, let's stop this, and you have to work on something else now. Just to fix it, and then come back. That reducing that times a lot more easier. Yeah.

Interviewer1

And what do you think will be the harm of using such a tool?

Interviewee

Obviously, if it gives you the wrong report here, you might waste time, right? Yeah, that's obviously a risk every time right. Like, most of the time, right? Intuitively you like as an engineer, like somebody says, Oh, this is working doing this. If I do this, here, like okay, I'm gonna go check this place first, right, like this part of the code like you like automatically you kind of do that and You might be wasting time that don't realize it's the some other component or like, no, but that happens even with any anybody. Right? So, yeah. Okay,

Interviewer1

so have you ever use?

Interviewer2

And once again, part that I'm going to follow up on that? Let's say that the tool is accurate? Yeah, we have this magic wand and the tool is 100% accurate? It's not, but let's assume that it is 100% accurate? Do you foresee any other problems or harms down the line because of using such a tool?

Interviewee

The thing I can say is like, it's a skill, debug things shouldn't make people lazy about it. Right? You still want people to have enough? That's an important skill anyway. Right? Maybe the tool works all the time, then. Okay, maybe we don't need that skill. Right. It's almost like seatbelt riding. If you don't crash the car, you don't need a seatbelt kind of thing. But if people are always like, the tool always works, then you don't have to worry about debugging skills. But that's a question right? Like if people become lazy about it, and like they don't develop the skills? Yeah. See, for me, like in a startup world, a lot of times you have to do things quickly. It's important to even debug quickly. So that's the important thing.

Interviewer2

So with that being said, in this imperfect scenario, where say 80% of the times it detects and 20% of the times it gives the wrong answer. Do you think that is actually a better system than a perfect system where you're still getting people to debug and learn about debugging? Because they know that it may not be the answer?

Interviewee

Yeah, I think still a better system, I would say. Obviously, like when you give a choice thing, okay, just look at these three files or something people will start at that, anyway. But a lot of debugging is mostly like two things, right? It's just writing console logs and like seeing what is going on? What is the data you're getting? And the other one is like, you're stepping through code one by one. I guess you could say a third things like looking at old existing logs of what happened. So yeah, I mean, if, if it helps through that, like looking through all logs, like these are all like this monotonous steps in a lot of ways. But it's able to get you through it quickly. Obviously, it doesn't mean like, so the other thing is like, just because it says this is where the bug is, you may not know what the right fix is. Also. The reason I say it is like, let's say it's because a bug because the data looks different than what you expect, you should know what that new different data is. So you can rewrite your code for it. So in those scenarios, just saying, Okay, this is where the bug is, may not be enough context.

Interviewer2

So what would be helpful in that context for extra things? This is gonna ask later, but I'm asking. So like, what else do you think would be useful along with saying this is the file to fix? And

Interviewee

I mean, saying, I guess liquids, like a function level, like even something like the parameters coming into the function. That might be good enough, right? Like, that might help saying, Oh, this is what the kind of data that comes here, like, because every function is like a contract saying, this is the kind of data you give me. And this is what is the candidate, I'm gonna return back. If the contract of the input data is different, so it kind of like breaks things. So that is like saying, Okay, this probably like this function, or like these functions where the error is just using Stack. If you actually run the scenario and say, Okay, there's a way to define saying this function expects this kind of data and like this given a different kind of data. That's why the bug is that might be fascinated by like. So

So you're saying more like reproducibility rather than automated reproducibility.

Yeah, there are mostly in big complicated systems when you have like bugs which are impossible to reproduce. Like, in sometimes in Windows, right, we'll have like, like, the main thing will be running like 30/40 threads. It just depends some specific timing. If you put a debugger on a thread right now, that will slow down that thread where the timing goes off. So those things happen, which are way harder like that's, in that case. It's mostly developer, like guessing things, going through code and like guessing what's going on. But that's a very, very rare scenario. In today's age, I would say like, nobody's writing like that kind of operating system. Like in the server client world today, it's mostly straightforward things.

Interviewer2

And will you? Okay, I'll ask this question later, because it's going to come up.

Interviewer1

Okay, so have you ever used any kind of automatic tool in your debugging process?

Interviewee

To so the kind of tools we use back in "complex system" was like, we used to have this thing called time tracer, where you pour as the bug, right, like the kind of scenarios or time or like threading or whatever, right? We they stopped this thing, what it does, like it will, every instruction ran in the CPU it will save the register information, and the assembly line or whatever. And it will, like, create this massive tracing file. And as an engineer, I could just go back, run through the trace, put breakpoints at the same point, like see, this thread is doing this, then this thread is doing this. Okay, this is what is causing it. Yeah, that was probably the most useful tool in that scenario I've seen. Other than that, like nothing out of ordinary, like I said, console logs and like locally, reproducing it with breakpoints and stepping through code, so.

Interviewer2

And console logs, how do you look for what you want in console logs?

Interviewee

I mean, unless you lock the data, whatever I like, most of the times, you're just saying like, Oh, it reached this point.

Interviewer2

Yeah. I mean, do you use like, automated tool? Or do you just do Command F, like grep or something?

Interviewee

It's almost like so in the current product, it was like Google Cloud, they like their functions has like logging. It's like almost console dot log, you can replace printf kind of thing. It will just, they'll have a huge log of it. And you just go and see what is going on there. And we also like anytime there's an error you, we send a message to our own, like messaging system, like Slack, and like, say, this error happened. So we can go and see if it's fine, or

Interviewer2

whatever, the log files, so the log files are not very big.

Interviewee

You can like filter it out to certain, like function instances, that's like a request comes, you can say, okay, just for that, give me all the logs. So you have like some fine tuning like that. Okay,

Interviewer1

so yeah,

Interviewee

one more thing we do is like, we have like a global catch, try catch kind of thing. So anything that comes like we are globally catching any error, and like, we are putting the exception like the stack and everything. That's a little bit of useful thing. So,

Interviewer1

we have already discussed with you about the our definition of bug localization tools, so according to your perception, what are the minimum required functionality? Do you think a bug localization tools should have

Interviewee

I mean, just saying file, this file, and that file may not be enough. Because not everybody splits or like makes file smaller. Like I guess like just identifying like, possible three, four functions within a certain class or something that will be useful. And like I said earlier, like, there's a way to like, reproduce the bug. And your tool is able to trace route and say, Okay, this function, like, this is the kind of data it expects, but it got this data. Right? Maybe so you You're not handling it correctly, the error can be two place, right? Like, either the color is the mistake, or the function didn't expect it and it's handling it incorrectly. So just knowing that will help figure out okay, where to go and fix things. Okay.

Interviewer1

So our next question is like, what do you think is the lowest acceptable performance of a bug localization tool? Like in terms of accuracy, or any kind of thing like currently, that tool, any tool can return Like top five potentially buggy files with 90% accuracy or 10 files with 100%? Equals like, what is the maximum number of files do you think is too much.

Interviewee

That's very dependent on the project size. Okay? Like, like, something like a big like "complex system" just saying, okay, within this team, or something is good enough to say, Okay, this team, whatever feature you're working on, that's where the bugcomes from. But like smaller startup, like, Yeah, three, five files, that will be good.

Interviewer2

And what kind of accuracy would be good, like, acceptable?

Interviewee

I mean, anything or sound, the person is pretty good?

Interviewer2

What kind of accuracy Do you have, as a developer? When you start looking right, like, how many times do you get the right place?

Interviewee

I mean, so from my debugging is one of my skills, right? Because generally, like even during "complex system" days, like I'll be told, like this bug, then I'll immediately know which place it is. Like, if people just tell me I did this, this happened. I know immediately why it happened. Because I'm kind of like, have memorized the code in my head. So I can immediately go and figure it out. It's only when there's the external data coming in where I don't know what it is, I'll have to go and step through things. Even other wrestling most of the times, kind of like more accurate. Take the worst case, if it takes little more time, start adding counselors everywhere and just see what's going on. I can do more for reproducible bugs, which is, which happens most of the time.

Interviewer2

And in your experience in managing teams, and like not people, but like being the architect and other people working on things. How often do you get people to come? How often do people come and ask you? Hey, I'm stuck in this bug and I don't know where to start looking at.

Interviewee

I guess it's a lot more like, what's the right fix?

Interviewer2

What's the right fix? Not where is the fix?

Interviewee

Like people kind of fire and forget this thing is doing this. Right? Like all these like deep, like console logging or whatever all those things most people do. It's mostly about what is the right fix? And sometimes it's very hard to decide. But that's, that's what is more discussed or like more I guess, creates more fights. Because different people think they had to do it differently. Okay, so

Interviewer1

what is your expectation of latency for such a bug localization? Like, do you expect that that tools should give output in five minutes? Or is it okay? If the tool gives output in one hour? Or maybe two hour?

Interviewee

Dependent on how urgent it is, right? I guess if it's urgent, most people may not even want this one to jump into debugging it. Yeah, obviously if it's five minutes people don't even do that. It's pretty fast. Honestly. One hour is pretty decent time to wait for because like, like buy a good fix. Yeah, like a day to fix anything on our is pretty big time to just paid for. Yeah.

Interviewer2

How long does it typically take you to find where the bug would be like, on average, or what was the last last bug that you fixed? Like how long did it take you to

Interviewee

I mean it's, I would say on average, it's probably like 3040 minutes. Again, depending on the size, I'm just saying like the major ones probably take 30 minutes to figure out What's going on? And probably another one hour to fix it. minor things, you probably can figure it out in like five, five minutes. Yeah. Okay. Okay, so

Interviewer1

our next question is like, do you think other kinds of output from a bug localization tools will be helpful? For example, like previous bug reports similar to the current one, or the best person who called fixing this current bug? Do you think this kind of output may help the developers?

Interviewee

Maybe, I guess if it's a distributed project, I'd like a GitHub. So other you can say, Okay, this guy should be the one fixing it. But like, in a company or within a team, it's like, it's probably more like a resource management, then, if somebody is too busy with something important, even though it's there, part of the code might get assigned to someone else, right? So those scenarios, it's very hard to say. Okay,

Interviewer1

and do you think, that will helps the developer, if we the tool can give any kind of like explanation why the tool thinks that particular file is responsible for that particular bug?

Interviewee

Yeah, like I said earlier, as we go, we say, Okay, this is the contract for each function, or like this class expects this kind of data and I, and things are outside the boundary that localize it enough to go by things faster.

Interviewer1

Yeah. Yeah, as far as like, reproducibility purpose, but do you think that junior developers can also learn from the explanation of the tool? How to debug it?

Interviewee

Yeah, I mean, obviously, it's a skill people have to learn, right? Like engineering, like, I guess, like, is just giving steps and everything would obviously be useful. But it's, like people have to keep doing it to learn more, learn more to get faster at it.

Interviewer2

So So I think one of the things that we're trying to get at is, instead of just telling where the bug is telling why this is a bug, or telling something about a fix by saying something this is similar to this bug, which was fixed this way, or something like that, would that be more helpful than telling list where it is?

Interviewee

Maybe, like, really junior engineers, it might help. Like, like an initial start, but I think everything has its own picks. And Glavine is also their own individuality on how they think. It's useful in early stage, but other people have their own style of doing things they probably won't even care for, if they know what exactly it is, they're gonna do it their way.

Interviewer2

Because we will be not me, but the state of the art uses machine learning. It basically just predicts a ranked list of files. Doesn't say why this files that rank list for this work? Why be helpful? Why we think these files are

Interviewee

that will be useful to just know. Yeah, so just as a data point, you want to like, does the wrong list? Right? Yeah. might be good to know why the tool thought in that way. Okay. So yeah,

Interviewer2

yes, of mistakes, like the tool giving the wrong answer. Knowing the why you may know that it's the wrong answer quicker.

Interviewee

Yeah, like next time, you probably see similar output, like maybe trying to think of this similar scenario. Yeah. Yeah.

Interviewer1

Okay, so who do you think that this tool will be useful for? Is it more suited for junior developers or for experienced developers security?

Interviewee

I mean, if it's accurate, like we said, like everybody will use it. Even seeing it as a time crunch, right? Like anytime you're releasing time for doing some task for anyone. So what like the additional like the way to fix it or all those things, maybe more junior engineer thing, like telling a senior engineer just what the issue is, then they can fix it. They'll know what to do. Okay.

Interviewer1

And in terms of roles, do you think the tool will be best suited for developers or it is best suited for like QA engineers?

Interviewee

I mean, it doesn't identify. It will be. It depends, right? Like the tool is something the developer has to run, and he does something. A lot of people would rather prefer the QA engineer, file the bug report with all the information. So developer doesn't spend more time and developer times or is considered more expensive. So in that way, maybe. But again, it's more a company process than honestly. Okay.

Interviewer1

And the next question is, what types of bugs would this be useful? For? Like, do you think that you should focus on the easy ones or the difficult ones that like includes multiple infrastructure, multiple machines, multiple deployment, those kinds of complex products? When

Interviewee

I would say like, simple as easy for junior, third layer, it will be useful. As the seniority goes, like, the more complex ones is where they spend more time on. So that just depends on the, like, if you get a senior person, like a simple bug to fix, they're gonna figure it out pretty fast. Like, tool may or may not be that useful.

Interviewer1

So where do you want output of such a tool? So is that good that in the issue management system? Or do you want it in the IDE of the developers?

Interviewee

I mean, like companies would want to track which files are like that. I mean, both will be good. Id obviously for just for the Dow first scenario, but like, generally big companies, like for 10, they want to track everything, right? Like they want to track. Okay, this feature this good part of the code base is like more problematic and a lot more bugs, like, I mean, like, you want to have both right? Like, we want to know that eventually where the fix was, and where the prediction was, and like, companies will want to track that in general. So

Interviewer1

would you use such a tool in your organization or in your practice?

Interviewee

If it's yeah, if it's fastening speed things up? Yeah. Okay,

Interviewer1

so the next question is, what do you pay for such a tool?

Interviewee

I'm not paying company pays for it. I'd

Interviewer2

like to argue for the company to pay for this.

Interviewee

I mean, if I think that I you know, engineers who are like Super Junior and like they take time for sure. Like, or like generally the kind of people who work with you feel like okay, they need time to do things. So medicine, try to use these tools, especially if you are in a very stressed kind of workplace. If things are like slow laid back, and like, everybody is good enough and doing their things, maybe not that much, right? It's very hard to justify also then. Again, like, I speak more on startup scenarios where I'm in. I mean, that's the mindset I'm in now so big companies may not it's something they might not even like, money is not a big problem for them. They're gonna only look at like, how much time are you saying overall? month or month or year or year?