Interviewer1

So I will start with the demographic questions first. So the first question is How old are you

Interviewee

40 years.

Interviewer1

Okay, and how do you identify your gender identity?

Interviewee

A male

Interviewer1

Okay, and what is your highest level of formal education?

Interviewee

Master's.

Interviewer1

Okay. And how many years of experience

Interviewer2

on second and in what program? Oh,

Interviewee

sure. Master's in Computer Science, especially computer networking architecture. Yeah.

Interviewer1

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

Interviewee

Okay, so now it's I should say maybe 16 years. Okay.

Interviewer1

And how many years of experience do you have been leading Teams?

Interviewee

Leading Teams? Yeah, maybe like 10 years? Yeah.

Interviewer1

Okay. So these are all our demographic question. Now we'll jump to the interview question.

Interviewee

Leading by leading teams is I'm not a manager. I'm technically leading with it.

Yes. Yeah. Technical Lead is still fine. We don't necessarily need like, as long as there's someone who reports to you on some kind of technical thing and you even if you're just a mentor, that's fine for us.

Yeah. Okay. Then it's more it's maybe say 14 years.

Interviewer1

Okay. Okay. So we start by interview questions. So the first question is, what is your perception if we use the word bug localization. So the question is, what do you understand by these words?

Interviewee

Can you repeat that again?

Interviewer2

So if we say when we say bug localization tool, what do you think that tool is doing?

Interviewee

Okay. So when I say bug localization, I'm essentially seeing this as a tool which is able to pinpoint or to some accuracy, tell you the lines of code or the functions or whatever where the bug is. So that's what I would understand by in these two words.

Interviewer1

Yeah. Okay, great. So in this study, we define the bug localization tool, a tool that can identify potentially buggy files given the bug report or stack trace, or both. Suppose the repository contains 700 source code files, and based on the bug report that was a shortlist a few files potentially responsible for the bug. The tools output is a ranked list of files sorted based on their probability of being responsible for a particular bug. Then the developers can start further investigation or maybe fix procedures from there. So that is our definition of back localization tool.

Interviewer2

And essentially we start from so it's very similar to what you were saying Kishore, just the difference being that the granularity may not be aligned, but maybe even a file or method or something like that. But it starts from a bug report. It's not when someone wants to debug and they know what bug it is or something like that, but more so when a bug report is filed. We try to find out where the fix for that bug report would be.

Interviewee

Understand. So given some input, which is a bug report, so this tool will be able to localize it to whatever granularity it is and to where the bug is. Yeah, we I have the same expectation.

Interviewer2

And the bug report can have like English language description, natural language description, our stack trace or anything like that.

Interviewee

Okay, that's interesting, because I didn't expect it to be natural language. I expected it to be some sort of log lines or some sort of, like stack. So I hoped for that but this is better.

Interviewer2

So those will also those may or may not be there but often bug reports natural language as well, right. And so with the natural language, if there is a log file or stack trace, and we use that as well, but if it is not there, then we only have the natural language description. Got it? Got it.

Interviewer1

Okay, so given the state of the art of a bug localization tool we have just defined would you use such a tool in your in your practice

Interviewee

this will be very helpful, like company like "a software company". Because many times the log lines or even the stack traces or, I would say, an outcome of a bug rather than identifying the bug by itself. So there's an exception, but really why that exception happened. Is is the cause, right? The really the cause is the bug. That would be very useful. Yeah. Able to, in fact, save time, right? If the tools can do with higher accuracy, higher level of accuracy, so less of false negatives, false positives in this case, then it basically ties saves the development time on developers time to learn to say what they need to focus their energy on.

Interviewer1

Yeah. So what do you think will be the harm of using such a tool in your practice? Do you think about any harm that can be for developers or for anything?

Interviewee

In terms of harm? I'm seeing this as a tool or aid for a developer. This is not a replacement for a software engineer. Right? And there's always a part of fixing the bug as well. It's not just identifying the bug. And does it does your tool go to the extent of fixing the bug? Or do you anticipate to build that?

Interviewer2

Yeah, so we don't actually provide the fix, we just say, this is where we think you should start debugging.

Interviewee

Okay, so this is a tool, which is a very good thing to have in our repertoire, like what when you're fixing bugs, or when you're when you're trying to analyze issues, in terms of harm is, I don't see particularly harmful in any way. But one point is if, if the tool is not up to the market, for example, there are many false positives or false negatives, right. So then the challenge is, the additional time spent by developers instead of it's like, the too much reliance on the software means you're on a goose chase, which will end up nowhere. So high accuracy is important for this tool. Because the whole point here is to reduce the developer's effort in ident, when localizing the bugs on their own, so that would be beneficial. And as as and when the benefits go down, then the cost goes up.

Interviewer1

So I have a follow up question here. So do you think that false positive and false negative are equally bad, or one is much worse than the other one?

Interviewee

So, as I said, it's really relying on this tool, right? When, for example, I'll give an example If GCC throws out some error, you tend to rely on that thing. Okay, I believe there's some error here, rather than questioning GCC is output by itself. Because we trust GCC, for example, in compilation to do the right thing, of course, the other GCC has worked, I wont say it doesn't have any bugs. But it's kind of paved path and, like Battle hunting tool. So we tried to rely on it, if suppose the bug localization is essentially pointing me to say, O this file or this line, or even this function has the bug. So you want to trust that tool, in order to reduce your effort in figuring out where the bug is, and the complexity of bugs sometimes, anyways, developers spend a lot of time. So that's what I mean by looking if it positive is this is where the bug is, if it's a false positive, then it's a waste of time to just not an add on and run that function. What is going on? The false negative is basically saying, Hey, don't look at these files that these functions are all good, then you're not paying attention to any of those. That's how I look at.

Interviewer2

Yeah, and do you think one is worse than the other like false positives, like sending you down the wrong path, or preventing you from looking at a particular path? They're one worse than the other, both of them are equally bad.

Interviewee

I think false negatives much worse than false positive, if you trust the tool, so being very sure for the negatives is quite important for a tool like this.

Interviewer2

Okay. So you would be okay with saying, let's say we give you five files to look at, and maybe the first four are wrong, and only the fifth one is correct. That's still, that's still better than giving you five files with none of them being?

Interviewee

Yeah, yeah. So high level of accuracy is more important than precision. Okay.

Interviewer2

Okay. And when you say high level, I think this is one of the questions that will come later. So I think we're talking about this, I'll bring this up. Do you have like a gut feeling of what kind of accuracy that you would like, the step I know of you, preferably, you'd have been 100%. Given that, like, what, like, you know, that, like static analysis tools have only about 20 30%? true positives, right? And so like, what kind of tolerance do you have for accuracy? Do you think?

Interviewee

So that's a good example, in case of static analysis tool, right? So we don't rely on static analysis tool at all. Okay, we do use it, that catches some of the very, very basic ones. And that's, that's only the reason why we use it. Okay. So example, we don't want our builds to break, or we don't have our lint errors to show up. So those sorts of things we do. But beyond that, we don't rely because we don't trust that software to catch the books as accurately as. So when you're building this tool, or in your case, it's a research to like look at this very early on, I gave an example of GCC, right? If GCC does, we try trying to test GCC saying, okay, it does accurate prediction of what is wrong in this? So, yeah, so that level of effectiveness, if you get from the tool, I think that will build trust in the tool itself or the purpose itself.

Interviewer2

Okay, so you would say, like 90% 99%, accuracy kind of things would be better?

Interviewee

Yeah, it would be ideal, I understand 100% is quite a tough task to get there. But as if you're looking anywhere between 80% and above accuracy, so that is great, that is really good. But the precision may not be there, precision may be like, Oh, these files are these five functions. And still, that reduces the job, it will narrow down the users to focus on certain things, then the whole code base altogether, potentially the whole I have, Path,

Interviewer2

on that note, let me ask you, like, related question is that, let's say that we can only do this accurately for 50% of the bugs. But for the remaining 50%, we can accurately say that we don't know

Interviewee

that that's a very good number to have, say, for example, you have 10 bug reports, for five times you say your prediction is with 80% accuracy. If you give the accuracy number, then there is an implicit expectation from the user of it the saying that, okay, maybe the tool is not able to figure out maybe on my own here, but at least they have that information, not having that information is much worse.

Interviewer2

And it but if I say for some of the bug reports, I rather say I don't know, then send you down the wrong path. Do you think you'd benefit that would be beneficial, that is

Interviewee

highly beneficial, the tool to be saying that? It doesn't have high confidence to tell where exactly the work is? Great. So it's better to not rely on the tool in those cases, rather than relying on the tool and going on the road down the wrong path? Yeah, perfect. Okay. Okay.

Interviewer1

So our next question is, have you ever used any kind of bug localization tool or debugging tools in your practice?

Interviewee

Not the bug localization tool? Not

Interviewer1

are any kind of automated tool for debugging? Sorry. Good. So have you used any kind of automatic tool for debugging?

Interviewee

Yeah, I mean, use many automated tools, right? The static analysis we have done in the past to figure out what it is. And many times even editors have some kind of plugins, although I don't remember what they are. But yeah, those things will let you kind of do some sort of analysis to figure out where exactly some problem lies. And now, now, recently, there are so many code helping tools by itself, when you're right, when you're coding, they help you a lot in terms of not having bugs in the first place. And those those are great, right? Sometimes a lot of debugging is of course, GDB, those sort of debugging tools are common to be used.

Interviewer2

Like, can you give us an example of this code helping tool you may have used recently have, you know, I don't know

Interviewee

what those plugins mean, for example, in IntelliJ IDEA, by itself has like many default ones, I guess, default settings to help you out when you're coding so that you don't make those mistakes. So okay.

Interviewer2

Okay, that that one's not like, not like something like a third party service or something that you're using that is doing cold suggestion or something like that.

Interviewee

Yeah, no, I'm not doing I know that Git has something right now select code generation tools. Yeah. Not those kinds of tools. I have not used those but maybe my team or someone else may be using it, but I have not particularly used those. So one thing I wanted to say for example, this is probably old school like rationalist you have many of these tools like Rational Purifiers, which used to catch bugs really good. Okay. And back in the day. It's very heavily used in my development career. Yeah. Perfect.

Interviewer1

So, you said that you use some of the tools from IntelliJ IDEA. So though, which function of that tool do you like most? Do you have any specific things that you like? The tool

Interviewee

Yeah, especially I pay a lot of focus on even warnings, many developers, I cannot say they do that, especially when when they start very early on, "a software company" started developing kernel code. So very pay a lot of attention not just to errors, but even to warnings, and also write a lot of defensive code. So my experience is certainly different than the New Age developers, I would say. So when I write code in IntelliJ, I don't like to have any warnings, even warnings in my code. So I like those tools, which are preemptively telling me that where are the potential errors that may arise? So fixing them as I'm coding is a very best way for me like not to introduce bugs in the first place. They're not logical bugs, which of course, the intellij cannot catch them. But syntax, bugs are, for example, if I'm mad me talk about Java. But if you C if you don't typecast in C or C++ accurately, then that's a very big recipe for disaster. So yeah, the modern languages or even like Java and Python have not about Python, at least, Java at least, there's lots of safety built into that. But I do program in C++ these days. So not that safety that don't exist as much as unlike Java or Kotlin. Go or other places. So yeah, particularly a good idea. I think I told you that this is the things I really like about the IDE integrate development environment, I also keep my settings. So the tolerance is low for such errors. So more and more idea, there is different levels of settings you can do right? So I keep my tolerance for errors or warnings very low, so that they can fixed. Yeah.

Interviewer1

So do you have anything any behavior of that tool that you dislike most?

Interviewee

So when all these features will make the tool bulky. And the IDES these days are like so bulky and CPU intensive. That I cannot open more than like three or two or three projects on my MacBook on a Intel platform, when Apple silicon is still better. But it is now not that great when it comes to Mac. Of course, if you have beefier machines or beefier laptops, this can work. The real problem I have is about the performance. And accuracy, right? Sometimes these tools are accurate, sometimes they're not. So that's one of the reasons why I'm talking about accuracy. So I have the accuracy. Knowing the accuracy, both it's quite important to trust these tools. Okay.

Interviewer1

So we have already shared the definition of the bag localization tool. So currently, those kinds of tool returns only the files that may be potentially responsible for a particular bag. So do you think is that enough? Or do you think any other kind of functionality of a bug localization tool may help the developers?

Interviewee

We spoke about the tool running given a bug report. And we spoke about providing this accuracy information to the users for each book. That is that's a quite vital stat to provide.

And again, can we run this in a batch mode that is becomes an important instead of being an aid to an individual developer? Can we run this? Yeah, this entire tool on a corpus of like bug reports. And can you have this tool given a bug fix you made? I mean, if you look at existing repositories out there, can you take all the bug reports that were there and run this tool so that it produces its own output and match Takens? What was the real fix altogether? So that is a good comparison coming out of this tool, saying that how and it's also a feedback for you itself, like how this is going.

Interviewer2

Yeah, so see you saying a feature like telling what was a similar bug report or a similar bug fix might be useful.

Interviewee

Similar is very useful, but also to I think that's a good point that if you have a bug, it carries tell it how similar it is to the other books that are here. They're there in the bug report. That's the batch mode, like running. So you have, instead of working in isolation with one bug report, you're working with a corpus of bug reports and to understand where they are. That's one of the common things we see as well, because the bug may manifest in many different ways. When you look at the log lines or something, that's because they because various inputs will trigger different paths. So if you can say that, Oh, this particular bug is the same as seen this is via the reports and other reports. So you don't need to go look at them. You, you basically do all those bug reports into one and solve one of them. Okay, because today, a lot of time developers the spend time is in scrubbing these bugs, saying to identify what they are. Yeah.

Interviewer2

So finding a similar bug, finding where that bug was fixed, finding who fix that bug, and all of that kind of information could be helpful.

Interviewee

Yeah, it will be helpful. I mean, there are tools to do it today. I mean, if you look at the log lines get blamed will tell you what exactly who did that. And as part of what bug I suppose fixing what bug they need is a new bugs. So that is, will be a report which is which becomes quite easy, right? Because for example, I go and I see some bug, I want to go see who fixed it and why they did it certain way. That gives me the context, whether I'm changing something which may go and break something else.

Interviewer2

So not just so you're saying not just saying what bug might be similar, but if we say that the bug is going to be in this file, also reporting the bugs that were fixed in that file in the past, giving that information will be helpful, because you may know why certain things were done.

Interviewee

Yeah, so one thing I want to like caution here is the granularity of a file has as benefit, but it's quite limited. I think the method or a code snippet is more useful rather than a file. So because some of the files we write tend to run like 3000 or 5000 lines, sometimes even like 12,000 lines of code. If you just point me to a file, that won't be enough, or I would suggest is just like how stack traces are. Because generally, if flow goes through some sort of functions in hierarchy. So in bug localization, if you can figure out the path is more path is more useful in the on the order of things in granularity. file is probably at the bottom. The next probably somewhere along the lines is a trace where it could be. And the more accurate one says like, method, then comes snippet. And the ideal is a line. The line. So this gradient, if you think of it, then you will see the usefulness of the tool.

Interviewer2

Yeah. Okay.

Interviewer1

Yeah, that's just a follow up question. You said that you want to understand why developers use a particular type of, so they think it would be helpful for developers, if that tool takes the developers why the tool thinks that a particular file is responsible for that particular bug.

Interviewee

Yeah, mean, for a particular bug report, if you say a particular file, maybe many methods are called in the same file. Generally, it so happens that I mean, if you look at C code, right, programming and see most of the functions are tend to be located in the same file. So then how do you triangulate which one to see?

Interviewer1

Yeah, so if it has some kind of explanations, say kind of that, that the bug report is talking about a particular function, and that particular function is in that file? That's why the tool is thinking about this file is responsible. So do you think this kind of explanation would help the developer?

Interviewee

Yeah, so at that point, you're identifying the function as well or a method as so which is more useful, rather than saying, because of this function, we are identifying this file. So the information is already there as function so versus okay.

Interviewer2

Yeah, I think one of the things some of these, like deep learning tools will tell you let's say we were going at the even at the line level, right? And it says, so some set of tools will tell you that this line is the where the fixes but on top of it, we can also kind of say Why the pool thinks this line is where

Interviewee

the bug fixes? Yeah. See, that's the additional information you're giving why it is this thing? So that is definitely beneficial. So can we let's look at it. If you have a bug, I'm just imagining one such book, there is a buffer overflow. So if you say that, okay, this bug report is, is originated are because of this function and this line, and the reason why it is a buffer overflow, or an integer overflow, whatever that is, so then you're basically spoon feeding the information to the user, which is i, which is much better. Okay, perfect.

Interviewer1

So another question is building that, that it will be useful for developers, if it if that goal points out the best person to fix a particular bug based on previous history or previous back fixing history.

Interviewee

Who's the best

Interviewer1

user to exact particular that?

Interviewee

I don't particularly see that as useful. But maybe in open source code base, it may be useful, not so much in companies. Because in a company setup, there is a team, generally a component, or even in that specific areas within that component was already kind of gun ownership. And generally, though, that team builds, say, for example, you have based on the history of property who fixed that line, the person may not be there in the company anymore. So yeah, there's minimal use, according to me in a company, but an open source, definitely. Yeah. Yeah.

Interviewer1

So the next question is about performance of what do you think is the lowest acceptable performance in terms of the size of the result? So do you think in that if we return the five top five responsible file, it would be good? Or should we reduce the size of the set? That's size and the accuracy is actually relatable to each other like in? If we return top 10, we can be 100% accurate, but if we return top five, then the accuracy may drop. So what do you think is the maximum set size that you can allow?

Interviewee

I would any day prefer accuracy over the size, but you should also control the size it cannot explode. So again, this is a kind of a is a kind of a call that you need to make based on what users are giving you feedback. Say, for example, you say you're 90% accurate with by showing like 10 files, versus you're 80% accurate by showing two files. So which is better? 10 is a little more, if you were to say five files, yeah, probably great, you should probably put a threshold of saying you won't go beyond this. And he will provide the accuracy number to that file, number of outputs that you provide, that is probably a better approach. And give it the flexibility for users to tune it, or set the number, that they're okay to take 10 results with 90% or 95% accuracy, or they want the lowest number of results with low number of accuracy. If you think about it, it is something that you got to hone the tool as well as try it with the feedback you want and get it the first time. But if you're thinking this is the production using this tool, right, then it that becomes much more important. If other than if you're just doing it for research and then initial like versions.

Interviewer1

Yeah. So another is in terms of time, so do think what should be what should be the latency of that tool do you think that tool should give output in maybe five minutes, or maybe one hour is also good for you.

Interviewee

For a batch which can be longer. For an interactive tool, it has to be shorter. If I get a bug report and then run this code analysis, there are many dimensions or factors to it, right? The the code base, how big it is, how many lines of code and then the complexity of the bug to identify where how the how there is an accuracy you need to generate. There's a lot of factors involved in that. I would say in the order of minutes is what probably you would expect on an interactive tool for a high level of accuracy, and for batch jobs, it can be much higher. Okay.

Interviewer1

So what do you think this tool will be useful So do you think it will be useful for newcomers or for experienced people

Interviewee

this will be useful for both. For example, I can think of this as a team making it a standard in the tool set. But any any bug reports come new. The it's automatically the tool is run just like we do see CI/CD, and you have against that bug report where that you localizing the bug. So whether someone chooses to use it or not, it's already there. So team would end up using it that way, rather than individuals making, are trying to do this on a needed basis. Yeah.

Interviewer2

So one of our follow up questions was going to be where do you think the output of this tool should be in an individual's IDE or on the bug repository? I think we know the answer.

Interviewee

Yeah, so that would be the way rather than on demand firing up this tool and running it. Some people may choose to do it. But I would think the majority would be an automated way of doing it. We can say for example, even now, previously, building some software was a explicit command or intent. Right? Now, as soon as you put something into the code base, you just run that CI CD, right? Whether we use the bits or not, it's secondary question. Yeah.

Interviewer1

So what types of bugs do you think this tool will be useful for? So do you think it is for easy bugs? Maybe some UI bugs? Or do you think it would be more useful for difficult bugs maybe related to multiple infrastructure, multiple machines or platform?

Interviewee

So when there is a bug report, especially, I can imagine that we heavy reliance on catching the low hanging fruit with this tool, okay, so and, and that's where it can make a major impact. Because in a developer's time, it doesn't matter. It's a small bug or a complicated bug, the number of hours to spend, matters. So if it's a small bug is spent one hour to fix that bug has more benefit versus the complicated bug. And if you took that same hour to analyze it. So if you can eliminate a low hanging fruit, especially in this case, that case a lot more value.

Interviewer1

Do you think that the tool will be best fitted for developers or it is for the QA engineers.

Interviewee

QA engineers, responsibility is about identifying the bugs, but not really the line of code. As I see, if there's a clean a clear delineation between QA and developer, many companies, that line is merging already. So I would say more about developers rather than QA.

Interviewer1

So the final question is, would you use such a tool in your practice?

Interviewee

Yeah, I think the answer is there is definitely use like when "a software company" right? So we, in "a software company", if you look at it, there is a developer productivity organization by itself. The whole idea for developer productivity is to initially increase developer productivity, how can you take over some of the developers tasks and automate them so that they don't need to spend time they focus on like software building or design and stuff. So we have a lot of tools in our repository here, which will help develop this for example, one is about I mean, some of the common tools you may come across is, is like metrics collection. So in our code, we write, okay, this is a metric I want to like collect. That's it, and everything else is handled automatically within our own framework that periodically it collects, it takes the values that it is a it's a counter or a gauge or histogram or whatever, and then publishes it to the metrics service and alerting you configure alerts, alerts get fired. And even in develop in development, a lot of these things are made annotations within Java, which does really heavy lifting, and code generation. Say for example, if you want say for example, we have a cloud tool where you can set certain properties for a service in runtime to change as soon as you change some configuration in the configuration system. Code will take that configuration Within a matter of seconds, and change the runtime system, according to the new value. So all these things, if you want to write code for it, it's going to be a lot of code. And it is just accomplished by just saying an annotating that saying, Oh, this is a dynamic property. And that's done. So for a company like "a software company", where we spend a lot of time to reduce the burden of developers. So this is very useful to. Yeah.

Interviewer1

Yeah. So the next question is, as you are a senior developer, so would you pitch to a company for paying for such a tool, if that is available for production use?

Interviewee

Yeah, why not? Right? So if this is a paid tool, definitely company will use it. Again, the cost and benefit is a analysis equal to do plus how much time it saves versus how much you're investing in this? Yeah, definitely.

Interviewer2

On that note, like what I mean, and this is something very subjective again, there's two kinds of models, right? Like one is that you have paying for a product, or you're paying for correct answer, like, because we know where you are already. you fixed the bug, because we have access to the repository. For every time we get it, right, we get paid when we don't get it right.

Interviewee

So that is a it's a kind of a tricky question, whether it is. Generally, the preference I haven't seen in that space, getting paid for a task that's completed, especially in the developer productivity side, it's been around like you purchase a license for a tool, and then the tool would run. But essentially, if you're pitching this as, hey, if you're using AI, to you're employing AI, let's say that you're employing AI as a developer, to go do software development or fixing bugs in this case, then would you pay for the developer time? I haven't seen that penetration yet. But hey, I can be wrong. That is where probably the market is moving. But so far, what I've seen is a licensing approach.

Interviewer2

Right? Yeah. Thank you so much, "". Yeah.

Interviewer1

So one last question. So that kind of tool would require access to a code base and all of those things. So do you think that would be a problem for company for, let's say about your organization?

Interviewee

Right, as I said, like, if the tool is running locally, that's no challenge because no information is going out. But if you want to run this like a cloud tool, then that becomes a challenge, I can see many companies like not wanting to do that. So one is you got to provide access to a code repository, which is generally behind a firewall provider. Or if you even have agents, which can run locally, then you're essentially letting some agent pass the code information outside, right? It's the via come a long way, trusting cloud services, let's say the past 5/10 to 15 years. Maybe a few years down the line, that won't be a matter at all. Do it some services we trust more, for example, "a software company" is mostly in AWS, and we use a lot of AWS services. But still, our code is not something that we are exposed to AWS services to the get an analysis. And neither does we don't expose it to like, for example, GitHub or anything. So we do have an instance of Git running within we run it as a service locally. Okay. The level of trust, yes, I can see that. We didn't trust cloud back then. We didn't trust to put data there. And that has changed over time. So even the federal government is putting data on a cloud so. So over time, that may go away, but maybe we are not there yet. Completely. Yeah. Perfect.

Interviewer1

Thank you so much.

Interviewee

Thank you. Hopefully, I was able to answer your questions.

Interviewer1

Yeah. Thank you.