Hello. My name is Brittany and this is Yoonki. We are currently doing research on how to improve the usability of static analysis tools. As a part of our research, we are conducting a series of interviews to get feedback from programmers, developers and static analysis tool users. Thank you for taking time out of your schedule to let us interview you.

During the course of this interview, we may ask you especially provocative questions, such as suggesting what you've told us is inconsistent with (quote-unquote) “best practice.” We do not mean to to insult or offend you, but instead to try to make you think deeply about why you do what you do. Try not to take anything personal and answer as best you can; there are no right answers.

Questions and Short Responses

For the first part of the interview, we're going to ask you some questions about your experience using static analysis tools. For now, we would like to focus on your own personal experience and opinions as a developer. Later in the interview, you will get the chance to be creative and suggest improvements for static analysis tools. We are dedicating 20 minutes to this section so try and keep your answers short, sweet and to the point.

Experience Questions

- Can you tell me about your first experience with static analysis?

- How easy/difficult was your first experience? Can you remember what made you feel that way?

- Do you have experience working on a software team? Did they use static analysis tools?

→ if yes, did you find it useful? (In your opinion did the tool accommodate teamwork(communicating standards/rule sets to be used)?)

→ if no, do you think a team could benefit from using static analysis tools during development?

- When you first started working at Google, was static analysis a part of the development process?

→ Did you have any difficulties or frustrations with using the static analysis tools provided?

→ Did the process/tools used make it clear what kind of bugs are a priority for Google? If not, how did you come to understand what you should and shouldn't be looking for?

Usage Questions

- When was the last time you used a static analysis tool?

- Have you used any other tools besides the ones you've already mentioned (name them)?

- Do you have a specific role as a developer (if so, what is it?)?

→ Do you feel the tool(s) used benefit what you do as a developer?

- Do you use your own static analysis or style checking tool (outside of the Google process)?

→ What is your tool of choice? Why do you prefer this tool?

→ Is there anything about this tool that you don't like, or find difficult?

- Do you use a static analysis tool for all the software you write? If not, why do you use it for some and not others?

- Are there any aspects of using static analysis tools that you describe as painful or difficult?

- Have you ever consciously avoided using static analysis? (Example?) Why do you think you avoided using it?

- Do you use/prefer static analysis tools that (a) run continuously in your IDE, (b) run in your IDE when you tell them to run, (c) as a part of your build process or (d) other?

Other Questions

- What type of bugs do you feel are, or should be, the priority at Google? (Feel free to specify both if you feel the type that are priority shouldn't be)

→ Do you feel the tools being used are finding these types of warnings?

- How did you find out about the static analysis tool you use?

- Do you use any other methods of debugging/finding errors in your code? If so, why do you prefer your method to using a static analysis tool?

- Some have suggested that static analysis tools should find bugs as early as possible in the development cycle. What would you define as 'earliest'? (When would you like for a static analysis tool to notify you of bugs in your code?)

- What, in your opinion, are the critical characteristics of a good static analysis tool?

- Can you describe the process involving static analysis that code undergoes here at Google?

- What are your goals when using static analysis?

→ Do you feel the tools you use now fulfill these?

→ If not, what's missing?

Interactive Interview (current workflow)

During this portion of the interview we would like to see you in action. We have dedicated 20 minutes to this section as well. (** Maybe see if they can use tool on real code and fix a real defect**)

If the subject has agreed to use their own code/tool:
Can you go ahead and pull up some code? As you're doing this we're going to ask you to do some things and ask some questions pertaining to how you do things.

If subject did not want to use their own code:
Here is the code we told you we would bring...would you mind running the tool on the code? While you do this we are going to ask you some questions.

Feel free to explain what you're doing out loud so we can get a better understanding of your workflow/thought process.

- Do the tools you use undergo any sort of evaluation before or after integration into your process? What about the tools Google uses?

→ Does your methods in any way reflect what you have to do in the Google process?

- Now that you’ve run the tool and gotten your feedback, what would your next move(s)? (Do you look at all the warnings? some, but not others? Fix all?)

- Do you configure the settings of your tool from default settings? (if so, show me)

→ About how much configuring would you say you have to do before you’re satisfied?

- Does your static analysis tool or process help in assessing what to do about a warning?

→ On average, how long would you say it takes you to figure out what to do about the warnings reported from your static analysis tool(s)?

→ FindBugs is in the process of adding Quick Fixes to their tool. Do you feel this would be helpful? What about partial quick fixes? What kind of solutions would you be looking for?

- What would you consider an overwhelming number of warnings?

→ Have you ever had an overwhelming number of warnings come up? (If so, how did you react?)

- Are there any guidelines or methods in place for keeping track of warnings that are new versus warnings that are old but still showing up? Does your process report when bugs are fixed?

→ If not, would this sort of mechanism be beneficial?

Do you feel like you need a break before we begin the last portion of the interview?
(If yes, take 5 minute intermission)

Participatory Design (desired workflow)

Now that we've got you in the mindset of programming using static analysis, we would like to allow you to be creative through a design exercise. We are going to give you a blank sheet of paper and we would like for you to describe to us your ideal workflow using a static analysis tool and draw a simple sketch of how you want to interact with the tool.

Again, we would like to thank you for your time. Do you have any questions for us?
(If no) Do you mind if we keep your drawing for further analysis?