

## **Introduction**

I'm here to understand better how we can integrate multiple static analysis tools in a single interface, particularly about showing bugs found for a same codebase by different analysis tools, feedback to know bug fix is ongoing and traceability of bug fixing. This interview will take about 45 minutes, during which time we'll go through some questions. Throughout, I'd like you to treat me as if you're describing the situation to someone who isn't familiar with software development and using static analysis tools. I'm here to learn from you.

A couple of things before we start. To the extent possible, I will take your comments to be confidential. I will aggregate all the comments from several interviews I'm conducting so that your comments are not easily traced to you. If I quote you in my final report, I will do so without identifying your name or specific role. If there's anything you really don't want on the record, even if it's anonymized, please let me know that, too. Also, this interview is entirely voluntary on your part – if for any reason you want to stop, please let me know. We can end the interview at that point with no repercussions for you of any kind. I can also throw out anything you've told me until that point.

Do you have any questions for me? All right, then, let's proceed.

I shall start recording now as mentioned in consent form. This is just so that I don't miss anything – no one other than me and if necessary, my thesis advisors will have access to the recording. Thanks.

## **Pre-test Questionnaire:**

Q. How often do you do software development (i.e., coding)?

- Is it daily, weekly etc?

Q. Have you used static analysis tools?

- tool to find bugs in your code!

Q. What tools have you used?

- Is it IDE integrated tool or any other dedicated tools like FindBugs, PMD etc.

Q. Which is your favourite one?

Q. Why it is your favourite?

- Any correlation to its better user interface feature?

Assume you are working on a project and want to find bugs in your code. There are two tools linked to your codebase to have better coverage of vulnerabilities. Now let us walkthrough 3 main questions with respect to its user interface.

### **[ Research Question 1]**

The first one;

Q. How to display the results of the same codebase from different analysis tools?

(present the prototype one after other in random order, do cognitive walkthrough – think aloud)

### **Tasks:**

#### **Prototype 1:**

1. Filter the bugs related to different tools.
  - Observe all 10 tools results one after the other
2. Identify the common bug
3. Fix the common bug

#### **Prototype 2:**

1. Scroll to see the results until tool 6
2. Filter the bugs related to different tools separately
3. Find the bugs related to tool 1 and tool 3
4. Identify the common bug
5. Fix the common bug

#### **Prototype 3:**

1. Scroll to see all results ( 3 screens )
2. Find new bugs ( Trending button )
3. Find the bugs with more importance to get fixed ( Priority button )
4. Find the bugs related to tool 1

5. Find alternative way to see the tool 1 related bugs
6. Find common bug
7. Fix the common bug

(after presenting all related prototypes)

**Follow up:**

Q. How do you feel about the home screen i.e., with statistics and diagrams?

Q. Do you wish to see the statistical representing charts for individual tools separately or complete tools in one?

Q. Do you desire to have 'union' or 'intersection' results when selecting multiple tools through filter?

Q. Among the 3 solution ideas presented, which one do you feel convenient with?

- easy to use?

Q. Why?

Q. Rate: [ 0 be low, 5 be high ] – all designs

Q. Do you imagine anything better UI than these? – Yes/No

Q. If yes, what does it look like?

Q. Here we have seen usage of 2 tools. If the number of tools increases, which UI do you think scales better?

**[ Research Question 2]**

Let's switch to second main question;

Q. What feedback works to know that the bug fixing is on-going?

(present the prototype one after other in random order, do cognitive walkthrough)

**Task:**

- Observe what happens after clicking on 'Fixed' button.

(after presenting all related prototypes)

**Follow up:**

Q. Among the 3 solution ideas presented, which one do you feel convenient with?

- easy to use?

Q. Why?

Q. Rate: [ 0 be low, 10 be high ] – both designs

Q. Do you imagine anything better UI than these? – Yes/No

Q. If yes, what does it look like?

**[ Research Question 3]**

Let's switch to final main question;

Q. How to carry traceability of bug fixing?

( Explain what traceability in this scenario is.)

Here in this scenario traceability mean to see how each attempt / commit to fix a bug effects the metrics of the analysis tools. Then a revert option to easily change the codebase version as desired by user. This ensures some safety to prevent new bugs.

(present the prototype one after other in random order)

**Tasks:**

1. Notice 'Know More'
2. See what this 'Revert' is.
3. Select all tools to see complete metrics.

[ Understand the scenario to take decision to revert to what level if needed. ]

(after presenting prototype)

**Follow up:**

Q. How you feel this UI will address the issue?

Q. Is this convenient UI for you?

Q. Rate: [ 0 be low, 10 be high ]

Q. How would you like to use this UI, in your bug fix work flow?

Q. Do you imagine anything better UI than this? – Yes/No

Q. If yes, what does it look like?

## **Overall**

Q. Do you think onboard phase i.e., explain each screen when user starts up the UI will help you better understand the UI?

- Is it helpful / distractive in this scenario?

Q. What we have seen here so far is from 'bug fix' perspective.

- How do you feel about 'code view' perspective?

- Which one do you prefer?

Q. Regarding traceability, would you prefer to trace the changes in process of fixing a bug or see the bug and find out what change caused this bug?

## **Conclusion**

Thank you – those are all the questions I have for you. If anything, else occurs to you later, please don't hesitate to let me know by email. Do you have any questions? Thanks again! for your participation and helping me understand.