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)

(for first three prototypes – results perspective)

Scenario: Assume you as a Software Developer working on a project called "Alpha". On next working day, you are about to see analysis results from multiple tools and your primary intention is to make your code base bug free.

Task: Identify the common bug reported by tools and fix it [not technically but as a workflow].

Prototype [single list]:

Success Criteria:
User find the **Bug name – XSS_CONFIG** which is common by **clicking** on **'Select all'** tools filter. Next, be able to know the bug fix workflow and finally **click 'Fixed'**.

Additional: Demo the user interface by showing multiple screen for each tool and significance of icon.

Prototype 2 [separate list]:

Success Criteria:
User scrolls to see the results **until tool 6** and find the **Bug name – XSS_CONFIG** which is common. Next, be able to know the bug fix workflow and finally **click 'Fixed'**.

[why not user scroll until tool 10 – reason: there are no bugs reported by tool 7, 8, 9 and 10. It signifies real time scenario.]

Additional: Demo the user interface by showing the bugs related to different tools separately

the bugs related to tool 1 and tool 3

Prototype 3 [tags]:

Success Criteria:
User scrolls down **3 screens** to see the results and find the **Bug name – XSS_CONFIG** which is common. Next, be able to know the bug fix workflow and finally **click 'Fixed'**.

Additional: Demo the user interface by showing trending (new bugs) and priority (the bugs with more importance to get fixed) significance and also show alternative approaches to see bugs reported by tool1.

(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: [O be low, 10 be high] – all designs in comparison

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

Q. If yes, what does it look like?

Q. Which UI do you think scales better?

(for first three prototypes – code view perspective)

Scenario: Assume you are a developer working on project Alpha precisely on software package called 'scripts_module'. On your next working day morning opened your code editor to start your work.

Task: What is the bug being reported by your analysis tools? And how many tools reported it?

Prototype 4 [multiple icons]:

Success Criteria:
On two clicks with different tool icons know that the bug is 'Anti cross site scripting filter'. The answer for number of tools is 2.

Prototype 5 [single icon]:

Success Criteria:
On single click on bug icon know that the bug is 'Anti cross site scripting filter'. The answer for number of tools is 2.

(after presenting both prototypes)

Follow up:

Q. Among the 2 solution ideas presented, which one do you feel convenient with?

- easy to use?

Q. Why?

Q. Rate: [O be low, 10 be high] – in comparison

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

Q. If yes, what does it look like?

Q. 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)

(now on results perspective)

Prototype 1,2,3: [animated icon, progress bar, popup]

Scenario:

Assume that you worked on a bug and changed some code related to it.
Next, submitted bug fix code for analysis.

Task:

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

[no success criteria as designer has to demo the feedback effect]

(now on code view perspective)

Prototype 4,5: [alert, status]

Scenario:

After submitting code for analysis, instead of being on the bug results window, you moved on to next task in code editor.

Task:

- Observe the visuals with alert box and status bar with spinner as demo.

(after presenting all related prototypes)

Follow up:

Q. Among the 5 solution ideas presented, which one do you feel convenient with?

- easy to use?

Q. Why?

Q. Rate: [O be low, 10 be high] – in comparison

Q. Would you prefer in having multiple feedbacks?

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. This ensures some safety to prevent new bugs.)

Prototype 1 & 2:

(results perspective)

Scenario:

You have been working on some part of code to fix a bug in last few working days. Now you would like to see how the changes (commits) you made are affecting the analysis results of other tools.

(present the prototype one after other in random order)

Task:

Watch the results of all tools and decide on a best commit among last 3 to revert as to start from that point.

Success Criteria:

Prototype 1: Selecting the commit id – fse254

Prototype 2: Selecting the commit id – fgd547

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

(after presenting prototype)

Follow up:

Q. Among 2 solution ideas, which one do you feel convenient with your work flow?

Q. Rate: [O be low, 10 be high] - in comparison

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

Q. If yes, what does it look like?

Prototype 3 [Before/After]:

Scenario:

As usual you are working on a code editor and notice a bug reported. You remember you have worked on this part of code before to fix some other bug.

Task:

Find out why you changed the code before and plan to come up with better solutions that satisfies the tools reporting same part of code. In this case, what was the code line you changed earlier to fix the bug reported earlier?

Success Criteria:

It is found out that code line " **value = value.replaceAll(“”,””);** " is uncommented.

Follow up:

Q. Do you feel convenient by this UI with your workflow?

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

Q. If yes, what does it look like?

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.