[Introduction]

- ☐ consent form
- ☐ start recording

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 ten 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?

(do cognitive walkthrough – think aloud)

(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 tool 2 and tool 3.

Success Criteria:

User find the **Bug name – XSS_REQUEST** which is common by **clicking** on **'tool 1'** and **'tool 2'** in filters section.

(code view perspective)

(pg dn - transition: tool123 -> src)

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 at XSSFilter.java file by your analysis tools?
And how many tools reported it?

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.

[Research Question 2]

Let's switch to second main question;

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

(pg dn - transition: code xssf description -> code rw3)

(results perspective)

[animated icon, progress bar, popup]

(pg dn - transition: Code XSSF description -> Code RW 3)

(explain pending, animated icons and progress bar)

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.

(pg up – transition: click home for code view)

(code view perspective)

[status]

(pg up – transition: alert)

[alert]

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.

[no success criteria for RQ 2 tasks as designer must demo the feedback effects]

[Research Question 3]

Let’s switch to final main question;

(pg up – transition for results view home)

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.)

(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.

Task:
Decide on a best commit among last 3 to revert as to start your work from that point.

Success Criteria:

Selecting the commit id – fgd547

(code view perspective)

(transition: click first revert for code view)

[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 in **XSSRequestWrapper.java** file to fix the bug reported earlier?

Success Criteria:

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

[Feedback]

- ☐ SUS Form

[Conclusion]