Program Navigation Tool Evaluation

Pre-Study Questionnaire:
Send questionnaire to all potential participants to balance groups for experience?

Decide Participant’s Group:
The group the participant is in will affect the order of the tasks.

- Tool + Task 1 -> No Tool + Task 2
- Tool + Task 2 -> No Tool + Task 1
- No Tool + Task 1 -> Tool + Task 2
- No Tool + Task 2 -> Tool + Task 1
Intro:
Hi, my name is <Researcher’s Name>. Thank you for participating in this research study. We are studying tools that help developers navigate source code in their IDEs. We hope to learn about how developers interact with these tools. Today, I’ll ask you to use two different navigation tools in the Eclipse IDE. Throughout this study you will be examining code written in Java from iTrust, an open source online medical records application. After these two tasks, I will ask you a few questions about the tools you used.

Consent Form
To get started, please sign this consent form. It basically says that participation is voluntary, your information is protected and anonymized, and we are not going to do anything unethical. Note that we are going to use audio and screen recording software.

(Make sure Active Presenter screen recording is on)
Eclipse Tool Training:
Demonstrate: Call hierarchy, Mark Occurrences, and Open Declaration

edu.ncsu.csc.itrust.dao.mysql.ApptDAO.editAppt(ApptBean) Line 134

This is a method used to edit appointments. I’ll show you three tools....

Mark Occurrences

The first tool is called mark occurrences. When you click on a variable it highlights that variable wherever it occurs in the code. *click on appt* As you can see, appt is highlighted in the comment as well as down here in the code!

Call Hierarchy

The next tool is named call hierarchy. It tells you the callers and the callees of a method. To use this tool, right click on a method name and select call hierarchy from the menu. That opens the call hierarchy view. *Invoke on edit appt* This view shows which methods call this method. By toggling this button, you can see all the methods that this method calls.

Open Declaration

Finally, open declaration can be used to show where a method is defined. *demonstrate on getConnection*

*Give the participant the mouse* Take as much time as you need to familiarize yourself with these tools. Let me know if you have any questions.
Task 1:
This method is used to open and read the contents of SQL files. Using the tool/tools I just showed you, tell me if this method ever receives user-provided input.

For example:

If this method could get passed input entered by a user in a web form, then the answer is yes.

If this method only gets passed hardcoded parameters, then the answer is no.

*Participant does the task*

Record their answer to the question. On a scale from 0 – 100, how confident are you in that answer?
Our Tool Training:
Demonstrate: Our Tool

edu.ncsu.csc.itrust.dao.mysql.ApptDAO.editAppt(ApptBean) Line 134

This is a method used to edit appointments. The tool I’m going to show you can be used to trace data flow throughout the program.

When you click on a variable it highlights all the instances of the variable in the code. Sometimes those instances appear off-screen. The boxes you see at the top and the bottom of the screen are used to display those instances.

When a variable is used higher up in the source file, or gets passed in from a method in a different source file, then a link to that location appears in the top box.

When the variable is used later in the program, it will appear in the bottom box.

Clicking links in the top box will take you to locations where the variable you selected is modified.

Clicking links in the bottom box will take you to the locations where the variable you selected is used.
Task 2:
This method gets called when users submit a web form. The web form contains many fields. Let’s focus on the fields pertaining to a recipient’s first name (line 54). This method saves that parameter, along with several others, to a form object. Using the tool/tools I just demonstrated, tell me whether the recipient first name field gets validated before the form object is added to the database. (To Do – comment where the form gets sent to the database)

*Participant does the task*

Record their answer to the question. On a scale from 0 – 100, how confident are you in that answer?
Questionnaire:
Make clear questionnaire relates to our tool.

*Questionnaire on separate sheet*

Open Responses:
This is the semi-structured portion of the study. Use the questions here to guide you, but don’t feel obliged to ask all of them or only questions from these questions. The goal is to generate discussion with the participant. What did they like, what did they hate, and (to a lesser extent) what suggestions do they have.

**Match between system and the real world**
Did you feel like the tool used conventions (highlighting, colors, and underlining) that you were familiar with?

**Visibility of system status**
Do you feel like the tool gave you reasonable feedback about what was it was doing?

**User control and freedom**
Using the tool, did you feel like you had freedom to explore the program?

**Consistency and standards**
Did the tool’s interface seem consistent to you?

**Aesthetic and minimalist design**
Did the tool present you with too much information? Not enough? Just the right amount?

**Error prevention**

**Recognition rather than recall**

**Flexibility and efficiency of use**

**Help users recognize, diagnose, and recover from errors**

**Help and documentation**

**Demographic Survey**
Finally, could you please fill out this short demographic questionnaire?