

Usability Test Results Report

Context

Usability tests were performed for a new VSCode (Visual Studio Code) extension currently under development, which aims to integrate the Test-Runner program to the interface of the popular code editor. In total, data were obtained from a preliminary session with 1 user and then another formal session with 3 users.

For the tests, a prototype was created using the interface design tool "Figma", using its functionalities to create interactive prototypes. The prototype was uploaded to the Maze platform for monitoring the test, where tasks and objectives were established for subsequent analysis.

Metrics Results

Effectiveness Rate of Performing Tasks

For this metric we must take into account the number of tasks assigned and completed. In the test, 4 tasks were assigned to be completed with a total of 4 testers.

Measurement

The first task was successfully completed by 3 of the 4 participants. This is reflected in the Maze report, which indicates that 25% of the users did not complete the task 100%.

In the second task we had a similar result, as another participant did not successfully complete the objectives.

For the third task, the completion percentage was better, as all participants completed the objectives. This was reflected as 0% in the Maze unfinished mission parameter.

For the last task, the same satisfactory percentage of completed tasks was obtained. Similarly, the percentage of unfinished mission was 0%.

Applying the metric formula to each task, we can obtain the following.

Task 1: $(3/4) \times 100 = 75\%$.

Task 2: $(3/4) \times 100 = 75\%$.

Task 3: $(4/4) \times 100 = 100\%$.

Task 4: $(4/4) \times 100 = 100\%$.

Now for the overall average we can do the following:

Overall average: $(75+75+75+100+100+100) / 4 = 87.5\%$ task effectiveness.

Conclusion

With this result we can conclude that we are within the average completion rate, which is 78% according to a study by Jeff Sauro conducted with a total of 1200 tasks. However, it is prudent to note that, in order to have greater reliability in the test results, it is necessary to perform them with a larger sampling.

Number of User Errors

Measurement

Test 1:

Task 1: Error encountering the test-runner icon, did not understand the instruction.

Task 2: No errors or further difficulty.

Task 3: No errors or increased difficulty.

Task 4: No errors or increased difficulty occurred.

Test 2:

Task 1: Error encountering the test-runner icon, ended the task before performing the last step.

Task 2: Tried to type in search bar.

Task 3: Wrong button in installation process.

Task 4: No errors or further difficulty occurred.

Test 3:

Task 1:

- Attempted to search in another section.
- Error searching for the icon.

Task 2: Attempt to run the test runner prior to what is needed.

Task 3: No errors or increased difficulty.

Task 4: No errors or increased difficulty occurred.

Test 4:

Task 1: Error encountering the test-runner icon, did not understand the instruction.

Task 2: Completed the task without the last step.

Task 3: Error in the installation order, due to reading instructions.

Task 4: No errors or further difficulty.

Now let's group the errors by task:

Task 1:

We will consider for this task had a total of 5 errors in 4 runs:

Makes an average of $5/4 = 1.25$.

However, 3 of the 5 errors in the test were due to user read errors.

Task 2:

Had 3 errors out of 4 runs:

Makes an average of $3/4 = .75$.

Task 3:

Had 2 errors out of 4 runs:

Makes an average of $2/4 = .5$

Task 4:

No errors were found for this task: averaging 0:

If we perform the calculation of the overall average number of errors per task we obtain that it is $(1.25 + .75 + .5 + .5 + 0) / 4 = 0.625$ errors per task.

Conclusion

According to a study by Jeff Sauro, the average number of errors per task is 0.7, and 2 out of 3 users make an error. In our sample we can observe that out of 4 participants at least 2 made errors related to the prototype and all 4 had errors related to reading comprehension, a factor that is not entirely dependent on our equipment. With an average of 0.6, we are in the average, considering the prototype as "easy to use", however, we must consider how these syntactic errors influence the use of our tool.

Interface Satisfaction Level

Measurement

10 questions were asked and answered by 5 persons.

Question 1: I think I would like to use this system frequently.

$$(1+4+4+5+4) / 5 = 3.6$$

This reflects an average of 3.6 in the satisfaction of this aspect, with which we can note that the program is pleasant for the user.

Question 2: I found the system unnecessarily complex

$2+4+1+1+1+1+1 = 1.8$ which reflects that the majority of users consider that the program is not complex.

Question 3: I think the system is easy to use

$4+5+5+5+5+5+5 = 4.8$ where 5 is very easy, in this section the users consider that they could use the program without major problems.

Question 4: I think I would need the help of a technical person to be able to use this system.

$3+1+1+1+1+1+2 = 1.6$ this low average indicates that the users consider that they can operate the program by themselves.

Question 5: I found the various functions of this system to be well integrated.

$(1+2+4+5+5+4) / 5 = 3.2$ on average, users consider the system to be well implemented.

Question 6: I found the system to be too inconsistent.

$(3+1+1+1+1+1+2) / 5 = 1.6$ being a low average, it refers to the system being consistent in the users' opinion.

Question 7: I imagine that most people would learn to use this system very quickly.

$(3+5+5+5+5+5+4) / 5 = 4.4$ in this case this value represents a high rate of positive perception, as users feel it would be easy to learn to use.

Question 8: I found the system very cumbersome to use.

$(5+1+1+1+1+1+1+1) / 5 = 1.8$ this average reflects that the users' perception is that the program is not cumbersome.

Question 9: I felt very confident using the system.

$(4+5+4+5+5+5) / 5 = 4.6$ This average reflects that the program does not generate a feeling of insecurity or uncertainty for the user.

Question 10: I needed to learn a lot of things before I could start using this system.

$(2+1+2+1+1+1) / 5 = 1.4$ This low percentage reflects that it is not necessary to have a lot of previous knowledge to use the tool.

Conclusion

Using the metric's test tool to obtain the average. The final average based on the sus score was 81/100. And using the "Table. SUS Results." our final grade is A, that is acceptable and Excellent.

For the most part, a satisfactory response was obtained for the use of the tool. The results obtained per question are within the acceptable average to consider that the users are satisfied with the tool and did not have too many difficulties beyond those involving reading or writing instructions in the test.

Interface Simplicity Level

With the goal to identify the level of satisfaction interpreted by the user about the interface, a Single-Ease Questionnaire was used to determinate this metric.

Measurement

The following questions were made for each task tested:

1. Once finished the Task 1: Installation of the extension, and in general, how easy, or hard was it to finish the task?

$$(7+6+7+6) / 4 = 6.5$$

2. Once finished the Task 2: Execution of the Help command, and in general, how easy, or hard was it to finish the task?

$$(7+6+7+7) / 4 = 6.75$$

3. Once finished the Task 3: Installation of the Test-Runner, and in general, how easy, or hard was it to finish the task?

$$(6+5+6+7) / 4 = 6$$

4. Once finished the Task 4: Execution of tests, and in general, how easy, or hard was it to finish the task?

$$(7+5+7+7) / 4 = 6.5$$

Conclusion

Although the average provides a sense of how easy it was to perform the task, what is of most value is the relationship of the score the user provided to the actions taken during the test, which can be reviewed in the recordings.

Observations on Elements Subject to Review

Below, the observations made in the application for each task are described, following the elements of the prototype subject to review for each of the tasks (This is described in the document "Guide for the Application of Usability Tests")

Task 1: Extension Installation

Applicants to the usability tests were generally able to find the Test-Runner extension icon in the VSCode icon bar, although it took some effort to locate it. This may indicate that the Test-Runner icon selected for the prototype is not sufficiently intuitive and/or eye-catching. It is felt that this can be mitigated by indicating in the extension description an image describing what the icon should look like in the VSCode sidebar, which will appear once the extension is installed. However, a new design for the Test-Runner extension icon for VSCode is planned.

Task 2: Command Help Execution

All applicants were able to easily locate the top bar for entering VSCode commands, a task that the test team estimated would take some visual effort for test applicants to perform. Everyone understood the usefulness of entering the Test-Runner extension's Help command, and several even attempted to read the instructions to know what to do next.

Most of the participants managed to close the Help file with ease once they were done with a quick reading of the instructions, however, there was one applicant who failed to close the file and finished the task earlier than stipulated. This is attributed to the fact that the applicant did not fully read the instructions and thought he had completed all the required steps.

Task 3: Test-Runner Installation

All participants were able to find the "INSTALL TEST-RUNNER" section with relative ease. Similarly, all were able to follow the installation procedure and were able to understand it, however, they were slightly confused in the order of the installation process and were a bit redundant in their actions when repeatedly pressing the "Install"

button, or selecting the "Open Folder" option and exiting the folder selection pop-up screen. However, what may be a bit more striking is the fact that a couple of participants attempted to press the "Install" button without first choosing a folder to target for Test-Runner installation.

Task 4: Tests Execution

All participants were able to find the "RUN TEST-RUNNER" section easily, requiring only a couple of seconds to locate it visually. Similarly, all participants were able to locate the button to run each of the tests individually, and were able to visualize the change of status of the tests to "Passed" or "Failed". Finally, all participants were able to locate the area where the test results are displayed with ease.

List of Positive, Improvement and Negative Findings

According to the observations made during the application of the usability tests, and the analysis carried out subsequently and described in the previous sections, the findings related to usability are listed below, segmented by positive, negative and improvement findings:

Positive Findings

- Easy to find the Command Bar.
- Easy to identify the Help Command.
- Easy to close the Help File.
- Easy to find "Install Test-Runner" section.
- Easy to find "Run Test-Runner" section.
- The change of Test Status is Easy Identifiable.
- Easy to find the "Test Execution" button.

Negative Findings

- Difficult to Find the Extension Icon

Improvements Findings

- Confusion in the order of the installation steps for the Test-Runner