University of the People

Writing Assignment Unit 2

CS 2401 - Software Engineering 1

February 7 ,2023

Consider the following nonfunctional requirements and determine which of them can be verified and which cannot. Write acceptance tests for each requirement or explain why it is not testable.

According to Marsic (2012), **functional requirements** are what determine “the system’s expected behavior and the effects it should produce in the problem domain” (p. 75). And they are what determines the product’s main features.

Non-functional requirements are what “describe some quality characteristic that the system-to-be shall exhibit” (Marsic, 2012, p. 75).

From the non-functional system properties described in the textbook we can determine whether the following requirements can be verified or not.

a. “The user interface must be user-friendly and easy to use.”

This can be carried out by allowing the users or the testers to use the prototype of the system-to-be. And improvements can be made by allowing them to interact with the interface of the system-to-be and making them comment regarding their user experience.

b. “The number of mouse clicks the user needs to perform when navigating to any window of the system’s user interface must be less than 10.”

This test can be done as well by carefully recording and documenting the mouse clicks that it takes to successfully navigate the interface. Also taking the user stories, developers can make improvements by making the system easy to navigate and well rounded. This can also be tested out by observing the users interact with the interface and noting down if users are able to perform their desired tasks with the number of mouse clicks less than. If it is possible to do that, the system will pass the test, and if it does not pass, further improvements in regard to navigation is required.

c. “The user interface of the new system must be simple enough so that any user can use it with a minimum training.”

This can also be tested out by observing the users’ interaction with the system-to-be. We can test this out by showing the users how to navigate around the system, leaving them to navigate the system on their own. Then, users will be able to report back whether they are able to navigate the system easily or not. If most of the comments are negative, we have to make improvements in the accessibility and user friendliness of the system.

d. “The maximum latency from the moment the user clicks a hyperlink in a web page until the rendering of the new web page starts is 1 second over a broadband connection.”

This can also be tested out by timing whether it is possible to make the new web page start in 1 second. If not, we have to check whether our system is too heavy or if there are too many images. And this might point out that we need to eliminate some unnecessary images and add-ons.

e. “In case of failure, the system must be easy to recover and must suffer minimum loss of important data.”

I don’t think that this requirement is possible to test out because various user circumstances can make it difficult to fulfill this requirement. Data can be lost in some cases and depending on the situation, sometimes preventing the data loss might be impossible. One way we can utilize to prevent the data loss is by backing up the important data. And this responsibility falls mostly to the user.

References

Marsic, I. (2012). *Software engineering.* Rutgers Unversity. **http://www.ece.rutgers.edu/~marsic/books/SE/book-SE_marsic.pdf**.