

Software Engineering Project 3

Problem 1:

Testing is meant to show that a program does what it is intended to do. Why may testers not always know what a program is intended for?

Solution 1:

Use cases developed to identify system interactions can be used as a basis for system testing. The use case of the software and the sequence diagrams associated with the use case documents play an important role in testing software. When testers do not know the use case of the program, then they do not always know what the program is intended for.

Problem 2:

Some people argue that developers should not be involved in testing their own code but that all testing should be the responsibility of a separate team. Give arguments for and against testing by the developers themselves.

Draw diagrams showing a conceptual view and a process view of The architectures of the following systems: A ticket machine used by passengers at a railway station.

Solution 2:

Developers know what is going on in their own code. They are confident in their own code. In addition, sometimes it took too much time to understand someone else code and find the bugs inside it. Therefore, it would be less time and money to involved developers in testing their own code. When developers test their own code, they might consider errors and problems which is not necessary or needed and that cause a waste of time, in some cases may be a waste of money and affects the costs of the project. Also, might fix errors that cause bigger problems for the software. In some cases when developers focused on their own code, they miss the important issues and problems in the software.

Problem 3:

What are the benefits of involving users in release testing at an early stage in the testing process? Are there disadvantages to user involvement?

Solution 3:

Involving users in release testing at an early stage have many benefits in agile methods. It can help to understand the corrections that software needed at the beginning and early stages. This can help to save time and reduce costs. In addition, in the end, it can help the project develop better with fewer issues and better aspects and values by getting customers' feedback. In the end, it can lead to building a great relationship with users and customers. I think the disadvantages of user involvement are that sometimes the user's requirements can confuse the testers and make it too much for them at the early stages and make the quality of the system poor because of wrong testing. Moreover, maybe the customer's requirements are not matched in the early stages and the software needs more tests.