How to decide on software testing techniques?

David Houngninou

CS 390

Dr. Hwang

12 October 2008
Software testing evaluates the capability of a program or system and determines whether it meets the required results. It is a trade-off between budget, time and quality. There are several reasons to test a program such as validation, quality assurance, reliability or performance. However, a main concern in software engineering is the implementation of test cases to maximize the efficiency of the testing process. There are several mechanisms that help developers decide on how to test a system. The selected article named “What information is relevant when selecting software testing techniques?”[2] presents the results of developing a characterization scheme for test technique selection. This paper will discuss the essential information required when selecting the appropriate techniques for testing a software system. Then, it will enumerate some issues that developers may encounter during the operational phase of the system.

A reliable program should conform to the entire specified design requirement and perform as required. The testing process is intended to identify possible design defects in the software. However, discovering those defects may be difficult according to the level of complexity of a system. It is nearly impossible to detect all the malfunctions in a system; however a good testing process should identify the majority of these problems, especially the most critical ones. Some tests are just intended to validate that the system works for a specified number of test cases. In this situation, since the number of test cases is finite, the developer cannot immediately guarantee that the program will work under all circumstances. Therefore, the target is to find a failed test to prove that the software does not work.

Testing can serve multiple purposes such as reliability, security or accuracy of a system. However, the developer is responsible for choosing a suitable testing technique for his own program. As explained in the article: “… their decisions would be based on sound knowledge of the techniques, instead of perceptions, suppositions and assumptions.”[2] The more advanced the testing techniques are, the more the developer is confident about the quality of the software before delivery. The choice of practical testing techniques sometimes becomes controversial. Some developers assume that they should be less costly while some believe that they should be more efficient and easy to use. As a result, it becomes hard to decide which technique is the best. The article listed the researchers’ thoughts, the
analysis of the consumer needs and the analysis of the programmer knowledge as relevant information. However, gathering all of these information and viewpoints requires following certain rules. To get an accurate set of information, we need to build the knowledge base with people who have a minimum of experience in testing and software development. The customer needs to be specific about what he expects from the software. The programmer needs to be experienced in the corresponding area of testing. The three categories of information are organized in three schemes respectively, a theoretical scheme, a consumer scheme and a programmer scheme. The consumer scheme is designed on the basis of the opinions of different consumers. The programmer scheme is built by getting viewpoints from developers or scientific publications such as research papers.

There are some tradeoffs in deciding which techniques to apply under some circumstances. If a testing technique is too complex and the testing team has no experience on how to use it, additional time and budget may be required for training. As described in the article: “… if the person who is to apply the technique has no experience (or knowledge) of the technique at all and there is no time/money for training, another technique might be used.”[2] If the testing technique is time consuming and needs to generate a large set of test cases, it may delay the delivery time of the final product. If a testing is efficient but not affordable enough, the final product price may exceed the budget line provided by the customer. Therefore, in the decision making process, developers need to consider performance, budget, and also time efficiency. First, the testing team needs to identify the quality attributes of the software (correctness, reliability, usability, etc.). Next, developers should identify the techniques that will be used to test each attribute within the constraints listed above (performance, budget, and time efficiency). Finally they should check if the technique was successful on previous projects. Since no testing technique is entirely perfect, the developer should be aware of the strengths and disadvantages of the technique he is planning to use. This can be done by communicating with other developers and getting feedbacks from their previous experience. Thus, the current project can be compared to any others in which the same technique was used. All these background checks can help the developer to know if the technique works as expected, and record possible failures.
The developer should know what quality attribute the testing technique can evaluate the best. As an example, testing a system for usability is unlikely to be the same as testing it for security or performance. Some structural properties of the software system such as its size, programming language, testing scope, or development method, also may impose some constraints on the technique to choose. Some testing techniques may be effective on small sized systems but not on large size system. Some testing techniques may only be suitable for some specific programming languages. The scope of the testing technique should be clearly defined. The testing scope should contain the parts of the system affected by the test and may vary from a single function to an entire module.

Even though the developer has decided on the best technique available for a system, he can still encounter some other issues after the release and delivery of the system. Finding the best testing technique does not necessarily mean that the final product will be free of bugs and defects. Generally a management team sets a time frame for software development. Testing goes until the release time or delivery time of the product is reached. Due to time and cost constraints, software may not be tested exhaustively. At some point, the developer has to stop testing to release and ship the software at the due date to satisfy the customer needs [1]. If the developer extends the testing phase to increase the reliability of the software, it may cause delays in the release time and large additional costs. Therefore, software may still encounter some failures during the operational phase just because the testing phase has not been applied thoroughly.

Several mechanisms help developers decide on how to test a system. Software testing heavily depends on the tester’s knowledge and experience. There are various methods and techniques available for software testing, but they do not provide the same outcome. The developer’s task is to find the appropriate techniques to detect the maximum of possible defects in the software. No testing technique is entirely perfect. The larger the program, the harder it is to test. By defining testing scopes, the tester can focus of separate modules of the code independently. By assessing the quality attributes of the software (correctness, reliability, usability, etc.) that need to be tested, the developer can make an educated choice.
on the testing techniques to use. Unfortunately, there is always a trade-off between budget, time and quality in software development. These restrictions may constraint the developer to shorten the testing process of system.
References:
