Swansea University

Computer Science

Software Testing

Critical System, CS-M13

Author:*Mohamad* Khaleqi*, 808956* Lecturer: *Dr. Anton* Setzer

December 2014

Contents

1 Testing Overview

1.1 What is testing?

Testing is the process of evaluating a system and the components to check if they are working correctly or not. Basically, it means to check the system to find errors, gaps, bugs and missing requirement. According to ANSI/IEEE 1059 standard (Software Verification And Validation Plan) [1] [1]:

"Guidance in preparing Software Verification and Validation Plans (SVVPs) that comply with IEEE Std 1012-1986 are provided. IEEE Std 1012-1986 specifies the required content for an SVVP. This guide recommends approaches to Verification and Validation (V&V) planning. This guide does not present requirements beyond those stated in IEEE Std 1012-1986." , testing is a process of analysing a software to detect the difference between existing and required conditions.

1.2 Who does testing?

In IT industry, large companies have a team to perform the testing. They know the requirements and they are responsible for checking every thing related to evaluation of a system. Also developers do the testing which is called Unit Testing. Each company has different methods for people who test the software basis of their experience and knowledge like Software Tester, Software Quality Assurance Engineer, QA Analyst and so on. on the other hand it is not possible to test the software at any time during its cycle. In most cases, These professionals are testing the system:

- Software Tester
- Software Developer
- Project Lead/Manager
- End User

1.3 When to start testing?

As soon as testing starts, cost, time and rework will be reduced and the project will finish faster. However SDLC [2] testing can start even before coding and programming when requirements gathering. It also depends on

[1] `http://standards.ieee.org/findstds/standard/1059-1993.html`
[2] Software Development Life Cycle

model used. For example in Incremental model, testing begins at the very end of project and in Water fall model, testing is changing during the test and project. Testing will be done in some different types at every phase of SDLC. For example:

- During requirement gathering phase
- At the end of each part of software
- When programming of software is finished
- Final test at the end

Reviewing the design in order to change or improve design is also kind of testing. In addition, testing by developer during coding is another type of testing which is called Unit type [2].

1.4 When to stop testing?

Unlike start testing, there is no certain definition about when to stop testing because testing is not a process to be completely done. Following are some points that testing can be stop [3]:

- Testing Deadlines.
- Completion of test case execution.
- Completion of Functional and code coverage to a certain point.
- Bug rate falls below a certain level and no high priority bugs are identified.
- Management decision.

1.5 The definition of V&V

In terms of software testing these two definitions are confusing. According to Roger S. Pressman:
"Verification refers to the set of activities that ensure that software correctly implements a specific functions. Validation refers to a different set of activities that ensure that the software that has been built is traceable to customer requirement. " [4]

Verification:

- Done by developer.

- Take place first and check the codes.

Validation:

- Done by tester.
- Take place after verification and and check the whole software.

2 Software Testing

2.1 Testing and Debugging

2.1.1 Testing:

Testing means identifying bugs, errors and problems in the software without fixing them.

2.1.2 Debugging:

Debugging means finding, stopping and fixing problems and bugs. Developers who write the codes, are involving to correct the code in this part. Debugging is one of the part of White box [3] or Unit testing.

3 Types

3.1 Manual testing

Testing the software manually and without any script or automatic tools. Tester will be like End user and trying to check the software for any unexpected errors or bugs. There are some stages of manual testing. For example:

- System testing
- User acceptance testing
- Integration testing

It can be also in some cases just to explore the software in order to identify errors.

[3] Or Glass testing means checking code and logic of software

3.2 Automation testing

Automation test is when the tester writes scripts and test the software with another software. It test the software from loading to exiting. It increase the speed of test and improve accuracy and on the other hand reduce the cost of time and testing to compare with manual testing.

4 Methods

4.1 Black Box Testing

In Black box testing, testers are like end users. They do not have to have knowledge about the software and they can check the software with some random inputs which is not obvious if the inputs are working with software or not. They do not have an access to the source code and also they do not know any thing about the architecture of the application. They only work with user interface. The advantage of this test is the tester does not need any knowledge about source code.

4.2 White Box Testing

White box testing is also called Glass testing or open box testing. In this test, testers have to look deeply the codes and check every things abut the code. With checking all codes, they have to find errors or any problems inside the software. They should have good understanding about the source and behaviour. The advantage of this test is, because the testers have an access to the source code, testing process is easy and they can check every area of software and even optimise the code. On the other hand it needs skilled tester in order to work with source code and it increases costs.

4.3 Grey Box Testing

Gray box testing is test the software, with limit knowledge about that software. In this test, testers have an access to code and database. In other words this test is in the middle of Black box testing and and white box testing.

5 Levels of Software Testing

According to Sommerville: “Development testing includes all testing activities that are caried out by the team developing the system.” The tester of

the software is usually the programmer who developed the software, however it is not always the case. All of this development testing process depends on the software and the purpose of it. For example in critical system, there is much formal process and always there is a separate testing team who is responsible about testing and maintain the test result. [5] In terms of testing, they are separated in two parts:

1. Functional testing
    - Unit testing
    - Integration Testing
    - System Testing
    - Acceptance Testing
    - Regression Testing
2. Non-functional testing
    - Performance testing
    - Usability testing
    - Security testing
    - Portability testing

5.1 Functional testing

5.1.1 Unit testing

Unit testing is the first testing on the software and developers perform the testing at the end of each part with test data to check if the software works properly or not. However this is not the best solution but this test happens during programming and can evaluate some errors. It is impossible to find all bugs and errors in this test. [3]
In this test, all parts of program methods, classes, functions with different input type data will be checked. This process also happens when some part of code changes. Because this test is usually done by programmers during programming, for testing time issues they can use mock object to do the test faster. For example developer needs to test an object and this object is depending on other objects or methods which are not ready at that moment or in other example are depending to database which can make the test process slow, in this case, developer can use mock objects. Mock objects are objects with the same interface but simulating the functionalities of external

or original object. Example of mock object in term of database could be an array with some data or for specific time, it can be an object with returning that time. [5]

5.1.2 Integration testing

Integration testing[4] is checking if functions or objects are working correctly together or in other words if they are correctly integrated. There are two types of testing:

- Button-up integration testing.
  begin with Unit testing and then moving to higher level combinations of units called modules or builds.
- Top-Down integration testing.
  In this test, first higher level modules are tested and then low level module.

In developing software process, usually Button-up integration test happens first and then Top-Down integration test. [5]

5.1.3 System testing

System testing checks if the components and program's parts are integrated together and they do the right operation in the right time. In other words, it checks to see if the software meets quality standard. The definition is close to the Integration testing but according to Sommerville's Software engineering book, there are two main differences:
“

1. During system testing, reusable components that have been separately developed and off-the-shelf systems may integrated with newly developed components. The complete system is then tested.
2. Components developed by different team members or groups may be integrated at this stage. System testing is a collective rather than an individual process. In some companies, system testing may involve a separate testing team with no involvement from designers and programmers. [5]

”

This test is important because [3]:

[4] In some books it is called as Component testing .

- It is the first step in SDLC, where the whole software is tested.
- The testing environment is similar to the production environment.
- It gives an ability to check the software architecture and business requirements.

It is not easy to say how much system testing is needed and when it should stop. However most companies have some policy for this test. For example they check all menus which are accessible by user at least once and they check the user input areas with correct and incorrect data.

5.1.4 Acceptance testing

This test can be the most important type of testing as it is conducted by the QA team[5]. They test the software by some pre-written test case. This test is not checking the simple spelling mistake or interface gaps. It also check any bug as result of system crashers or even major errors in the software. By doing this test, the testing team finds out when the application will be in production. [3]

5.1.5 Alpha test

This is the first stage of testing. Unit testing, system testing and integration testing, all of them known as alpha testing. and system testing when combined are known as Alpha testing. In Alpha testing these areas will be checked:

- Spelling errors
- Broken links

5.1.6 Beta test

If the Alpha test passes the test successfully, then it is time for Beta test. Beta test is also known as pre-release test. In market, software with beta version are in this stage and End users can join the test at this phase by following:

- Install, using the software and send feedback.
- input random values and check crashes and bugs.

[5]Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client's requirements

5.1.7 Regression testing

To fix an error or a bug, the code needs some changing and this change may affect other part of function and it may be a reason for another bug. To verify that changing in code has not resulted in other parts, is Regression testing. This test can be useful because of reducing the gaps when some changes happen in code and makes the whole testing process faster. [3]

5.2 Non-functional testing

This section is about testing the software in non-functional parts. It means testing the software from requirements that are not functional in nature but they are effective and important in the security and maintenance. Some of these tests are mentioned as follows:

5.2.1 Performance testing

This test is about looking for any issue related to performance, not bugs or errors. In some cases the Software works correctly and satisfy all the requirements but it needs to fix some technical performance issues. Here are some of performance issues: [3]

- Delay in network
- Data rendering
- Client side processing

This test is mandatory in terms of one of following aspects:

- Capacity
- Speed[6]
- Stability
- Scalability

[6]For example response time, data rendering and accessing

5.2.2 Usability testing

This test is checking the definition of usability in software point of view. According to Nielsen:[7]:
[6] "Usability is a quality attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process."
Usability is defined in 5 quality components: learnability, efficiency, memorability, errors and satisfaction [6]. Molich [8] in Comparative evaluation of usability tests stated about these factors in each system should be present in order to be user friendly: easy to learn, remember and understand, satisfactory and efficient to use [7].
Also these methods and standards are about usability:

- ISO-9126 [9]
- ISO-9241-11
  "The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments." [8]
- ISO-13407 [10]
- IEEE std.610.12 [11]

In this term, UI testing [12] is about shape, color, size and other requirement but Usability testing is about to make user friendly software and also this test makes sure that the software is easy to use for end users. With this definition, UI testing can be a sup part of Usability testing.

5.2.3 Security testing

Security testing is about checking the software in order to find any gap or security bugs. There are some aspects that Security test should ensure:SQL injection attacks, Cross-site scripting attacks, Input checking and validation, security of software data, user privacy, authentication, confidentiality and so on. [3]

[7] Nielsen profile: `http://www.nngroup.com/people/jakob-nielsen/`
[8] Rolf Molich profile: `http://www.dialogdesign.dk/About_Rolf_Molich.htm`
[9] Full details are available at: `http://www.cse.dcu.ie/essiscope/sm2/9126ref.html` and `http://www.sqa.net/iso9126.html`
[10] Full details are available at: `http://www.usabilitynet.org/tools/13407stds.htm`
[11] Full details are available at: `http://standards.ieee.org/findstds/standard/610.12-1990.html`
[12] User Interface testing or Graphic User Interface testing

5.2.4 Portability testing

Portability testing is about to ensure that the software is working in other systems, is reusable. In this test, tester team build the executable file[13] and also to check if the software is working after transferring the installed file to another system or not. Operation Systems, Browsers and computer hardware are focusing in this test a lot because of purpose of their job. They should be compatible with other systems and environments.

[13]For example executable file in windows OS is .exe

References

[1] M. W. Hervey, "Software quality management recommendations," tech. rep., DTIC Document, 2006.

[2] W. C. Hetzel and B. Hetzel, *The complete guide to software testing*. QED Information Sciences Wellesley, MA, 1988.

[3] *Software Testing Tutorial*. Available at `http://www.tutorialspoint.com/software_testing/index.htm`.

[4] R. S. Pressman, *SOftware Engineering*. Mc Graw Hill, 2005.

[5] I. Sommerville, "Software engineering," 2009.

[6] J. Nielsen, *Usability 101: Introduction to Usability*. Available at `http://www.nngroup.com/articles/usability-101-introduction-to-usability/`.

[7] R. Molich, A. D. Thomsen, B. Karyukina, L. Schmidt, M. Ede, W. van Oel, and M. Arcuri, "Comparative evaluation of usability tests," in *CHI'99 extended abstracts on Human factors in computing systems*, pp. 83–84, ACM, 1999.

[8] T. Jokela, N. Iivari, J. Matero, and M. Karukka, "The standard of user-centered design and the standard definition of usability: analyzing iso 13407 against iso 9241-11," in *Proceedings of the Latin American conference on Human-computer interaction*, pp. 53–60, ACM, 2003.