

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 12, July 2017

Software Model Testing and Methodologies

Prem Vikas¹, Sachin.N¹, Dr.D.V. Ashoka²

U.G Student, Department of Information Science and Engineering, JSSATE, Bengaluru, Karnataka, India¹

Professor, Department of Information Science and Engineering, JSSATE, Bengaluru, Karnataka, India²

ABSTRACT: This paper reviews about importance's of software testing in IT field and the different methodologies through which testing can be achieved. One more criteria is about the evolution of testing which is to be not limited with error correction and rectification but also to meet the requirements of the product designer client specification, as meeting the requirements gives better chance of finding out all possible malfunction of the model(software). Also, the scope of testing can be viewed by various combination of inputs to check the robustness of the software. Thus, software testing is the parameter in the measure of the software development.

The purpose of this study is to present a systematic view of significance to field of testing and emphasising its existence in the current development of software MODELS.

KEYWORDS: System Under Test (SUT), Black box testing, White box of testing, Software testing, Testing life cycle.

I. INTRODUCTION

Software testing is a process of implementing a program or application with the intent of finding the bugs(errors).[1]. It can also be stated as the process of validating and verifying that a software program or application meets the technical requirements that reflects its design and development.

A primary purpose of testing is to perceive software flaws so that failures may be discovered and corrected. Testing cannot confirm that a product functions properly under all circumstances but can only authenticate that it does not function properly under specific conditions.[4].

Software testing has progressed since 1970's as a vital part of software development process. Through it, the final class of the software can be improved by discovering errors and faults through interrelating, checking behaviour and evaluating the System Under Test (SUT) to check whether it operates as expected or not on a finite number of test cases with the aim of determining errors that are found in the software and resolving them. [3].

Software testing is significant activity in Software Development Life Cycle(SDLC). Software testing is the process of computing the functionality and robustness of a program through execution or analysis. The testing of software is an important means of evaluating the software to determine its quality. Since testing typically consumes 40-50% of development efforts, and consumes more effort for systems that require higher levels of reliability. Since software-testing process is a very exclusive process, complete testing is practically impossible and it is not acceptable to reduce testing effort by accepting quality reductions.[2]. Testing effort is often a major cost factor during software development. Many software organizations are spending up to 40% of their resources on testing. Therefore, an existing open problem is how to reduce testing effort without affecting the quality level of the final software.

As the amount of continence and advancement of existing systems grow, significant amount of testing will also be needed to verify systems after modification are made. Despite advances in formal methods and verification techniques, a system still needs to be tested before it is used. Testing remains the truly operative means to guarantee the quality of a software system of non-flimsy complexity, as well as one of the most sophisticated and least silent areas in software engineering. This paper further classifies into types of testing methods adopted for the betterment in improving the efficiency of testing and also how testing is carried out and its importance in software development.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 12, July 2017

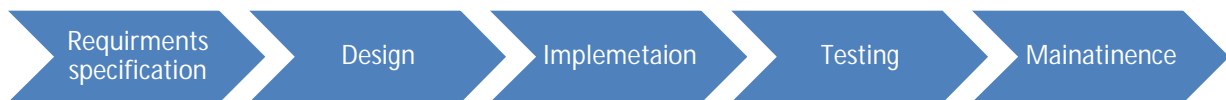
II. TESTING FUNDAMENTALS

WHO DOES TESTING?

It depends on the type of process and the associated team members of the projects. In the IT industry, large companies have a team with tasks to assess the developed software in context of the given requirements. Moreover, developers also perform testing which is called Unit Testing. In most cases, the following professionals are involved in testing a system within their respective volumes: Software Tester

- Software Developer
- Project Manager
- End User

Different companies have different terms for people who test the software on the basis of their experience and knowledge such as Software Tester, Software Quality Assurance Engineer, QA Analyst, etc. It is not possible to test the software at any time during its SDLC cycle. The next two sections state when testing should be started and when to end it during the SDLC.



Software Development Life Cycle(SDLC)

WHEN TO START TESTING

An early acceleration to testing minimizes the cost and time to rephrase and produce error-free software that is delivered to the end user(client). However, in Software Development Life Cycle (SDLC), testing can be started from the Requirements specification phase and continued till the integration of the software. It also depends on the software development model that is being used. For example, considering the Waterfall model, each step is carried out in top-down analysis fashion and testing is conducted in the testing(verification) phase; but in the incremental model, steps are carried out in increments i.e. each process is divided into small units and each units are developed and tested individually which results in testing performed at the end of every increment/iteration.

Testing is performed in different methods at every phase of SDLC:

- During the requirement specification phase, the study and validation of requirements refers to the path of testing.
- Reworking the design in the design phase intending to improve the design is also considered as testing.
- Testing achieved by a developer on completion of the code is also categorized as testing.

WHEN TO STOP TESTING?

It is tedious process to determine when to stop testing, as testing is a never-ending process and one cannot assure that a software is 100% tested. The following features are to be considered for stopping the testing process:

- Deadlines of project.
- Accomplishment of test case execution.

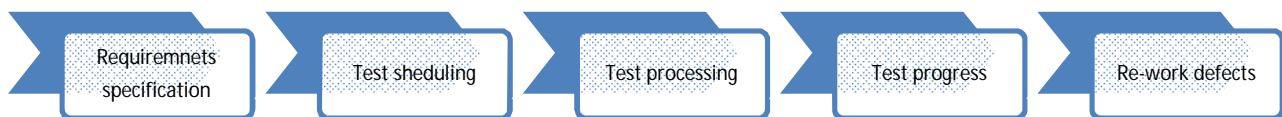
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 12, July 2017

- Completion of functional and code coverage to a certain point.
- Bug rate falls below a threshold level and no high-priority bugs are identified.
- According to Management decision.

SOFTWARE TESTING LIFE CYCLE



Requirement specification--Testing should begin in the requirements stage of the software development life cycle. During the design stage, testers work with developers in determining what aspects of a design are testable and with what constraints those tests work.

Test scheduling--test strategy, test design, test basic layout. A pool of activities are considered here to achieve test scheduling.

Test processing-- test cases, test functions, test background are developed to test software.

Test progress--Testers execute the software based on the test scheduling and tests process and report any errors found to the development team.

Re-working defects-- Once testing is completed, testers produce verification and make final reports on their test exertion and whether or not the software tested is ready for proclamation.

III.BASICS OF SOFTWARE TESTING

Basics of software testing falls into two classes:

- Black box testing
- White box testing

Black box testing is also identified as behavioural testing it doesn't bother about internal mechanism of the system but focuses on the final output with respect to the given input which means the tester cannot see the internal structure of the system it is also referred as functional testing.

This technique falls into error categories like:

- Interface errors
- Missing functions
- Initialization and termination errors

Steps carried in black box testing:

1. Initial prerequisites and specifications are observed.
2. tester plans combinational valid and invalid inputs to check the positivity of the software.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 12, July 2017

3. then proper constructs are used and test cases are developed.
4. after the execution of the test cases, the results are compared with actual results.
5. flaws if any are fixed and re-tested.

Types of black box testing

Prominent types are: 1. Functional testing, 2. Regression testing

Functional testing: it is carried out to check the functionality of the system works good as mentioned in the requirements. It also includes failure paths and boundary cases. It comes under black box testing.

Regression testing: it is carried out after making some changes in the product such as unit or a component to guarantee that the product is working perfectly after modification. The changes may include software enhancement or change in configuration. It comes under black box testing.

White box testing is also known as glass box testing it cares about internal mechanism of the system where the user can see into the internal structure of the system. It also makes use of programming skills of the user to design the test cases. It is also referred as structural testing.

This technique falls into coverage criteria:

- Control flow testing
- Data flow testing
- Decision coverage

Steps carried in white box testing:

1. realizing the source code-- important aspect of white box is to have a condensed knowledge about the source code implementations and its characteristics, which results in full exploitation of re-work of software. Also, the testing should be done with respect to the security criteria of the software.

2. developing test cases—since the development of application code is done by developer the testing in white box is also done by the developer. Here as on the coding is done simultaneously small errors are rectified and re-work is done and during testing process tests are developed for each process in application and verification is done.

Types of white box testing:

Prominent types: 1. Unit testing, 2. Integration testing

Unit testing: it is about testing the specific unit or group of interrelated units. A unit is a part of system and is tested to find errors and it checks each unit to analyse and fix errors and checks the functionality of the system.

Integration testing: it is about merging a group of modules to produce output. it comes under both black box testing and white box testing. It applies test cases to a module of that groups and produce output. It is sometimes called integration and testing (I&T).

Black box testing is done for performing validation and white box is done for verification.

Comparison of Black and White box of testing

Consider a function which performs a particular task which involves constructs and iterations.

White box testing involves in checking the robustness and structure of the function i.e. whether all the programming constructs are used correctly. It depends on the analysis of the constructs and its orders recommended by the design of

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 12, July 2017

the software and then the output of the model is verified. In case of black box testing it does not involve in the steps and constructs used but it just checks the correctness of the code with comparing the testing results with the actual results and also whether the software and hardware are in sync. As an example [6] of testing concepts, the following java code uses a Border Pane as a container for two Flow Panes and a Button. The first Flow Pane contains a Label and Choice Box, the second Flow Pane a Label and a List View. The Button switches the visibility of each Flow Pane.

Functional testing can easily be explained just providing the inputs by using GUI and the logic of the program how actually the programs works can be explained by means of white box testing concepts.

```
public class ApplicationWindow extends Application {  
  
    public static void main(String[] args) {  
        launch(args);  
    }  
  
    public void start(Stage primaryStage) {  
  
        primaryStage.setTitle("example Gui");  
  
        BorderPane componentLayout = new BorderPane();  
        componentLayout.setPadding(new Insets(20,0,20,20));  
  
        final FlowPane choicePane = new FlowPane();  
        choicePane.setHgap(100);  
        Label choiceLbl = new Label("Fruits");  
  
        ChoiceBox fruits = new ChoiceBox(FXCollections.observableArrayList("Asparagus", "Beans", "Broccoli",  
"Cabbage"  
    , "Carrot", "Celery", "Cucumber", "Leek", "Mushroom"  
    , "Pepper", "Radish", "Shallot", "Spinach", "Swede"  
    , "Turnip"));  
  
        choicePane.getChildren().add(choiceLbl);  
        choicePane.getChildren().add(fruits);  
        componentLayout.setTop(choicePane);  
        final FlowPane listPane = new FlowPane();  
        listPane.setHgap(100);  
        Label listLbl = new Label("Vegetables");  
  
        ListView vegetables = new ListView(FXCollections.observableArrayList("Apple", "Apricot", "Banana"  
    , "Cherry", "Date", "Kiwi", "Orange", "Pear", "Strawberry"));  
        listPane.getChildren().add(listLbl);  
        listPane.getChildren().add(vegetables);  
        listPane.setVisible(false);  
        componentLayout.setCenter(listPane);  
        Button vegFruitBut = new Button("Fruit or Veg");  
        vegFruitBut.setOnAction(new EventHandler() {  
  
            public void handle(ActionEvent event) {  
  
                choicePane.setVisible(!choicePane.isVisible());  
                listPane.setVisible(!listPane.isVisible());  
            }  
        }  
    }  
}
```

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 12, July 2017

```
    }  
});  
  
componentLayout.setBottom(vegFruitBut);  
  
Scene appScene = new Scene(componentLayout,500,500);  
  
primaryStage.setScene(appScene);  
primaryStage.show();  
}  
}
```

Thus, the white box testing implements the necessary statements and then checks for the results where as black box fails in doing so.

IV.CONCLUSION

For an organization, developing a software model that meets the business needs of clients is always a task. Software testing report for a large proportion of effort in the software development process, but we have lately begun to understand the restraints of systematic planning, execution and control. Due to the challenges faced by testing an overplus opportunities are developed in the field of technology which further helps in the betterment of the future of IT industry.

Realising the software test models, and types of testing methods and its importance which results in guarantee deliver of the software system to the client which is out of bugs and functioning according to the basic design without compromising the demands in outcomes results in complete function of software cycle (SLDC).thus, testing is integral part of Software development life cycle also an important factor which assure the quality, correctness,relaiabilty, dependence of the software.

REFERENCES

- [1] Software Testing. A Craftsman's Approach, Paul C. Jorgensen, third edition
- [2] Software Testing and Analysis, Mauro Pezze, Michal Young
- [3] Wikipedia.
- [4] Software Testing Techniques (Second Edition) by Boris Beizer.
- [5] Azhar blog on software testing methodologies.
- [6]<https://www.thoughtco.com>