Model Based Testing Process for Software Systems

G. Swathi, Girijamma H. A.

Abstract: ***Software Testing Process is a very significant issue that influences the standard of software system, that plays a very key role within the development of entire software system life cycle. Software testing is evolving, and Model Based Testing (MBT) is an integral piece of modern test automation. Compare with ancient testing strategies, Model Based Testing is in a position to maintain and achieve testing responsibilities in a quicker, inexpensive and very effective manner. MBT has grown interest with the familiarization of models in the software system design process and implementation process. This paper provides a outline of Model Based Testing and describes its approaches. It discusses software testing evolution. The MBT process is represented, and also the activities are discussed in detail. Additionally, challenges, benefits and drawbacks with Model Based Testing are briefly bestowed. It also describes the suitable applications of Model Based Testing.***

Keywords: ***System Under Test(SUT), Model Based Testing(MBT).***

I. INTRODUCTION

The Model Based Testing (MBT) is a software testing technique, which is used by testing team to check run time performance of software system under test against predictions created by a proper specification model. Figure 1. depicts the MBT framework. Model could be a representation of a software system behavior. Behavior will be represented in terms of actions, conditions, sequence of inputs, outputs and the information flow from input to output. Model symbolize the system that is being tested, its atmosphere, or the test suit itself, that straight away assist test plan, test analyzing, test managing, test development, test run and test reporting activities of software testing process. Model Based Testing explains systems behavior in response to an action. Several analysis are exposed that MBT is effectual, primarily when wont to test tiny applications, state rich systems, embedded computing systems, and user interfaces amid much advanced data. Model of software system could facilitate purifying ambiguous and badly outlined software system requirements.

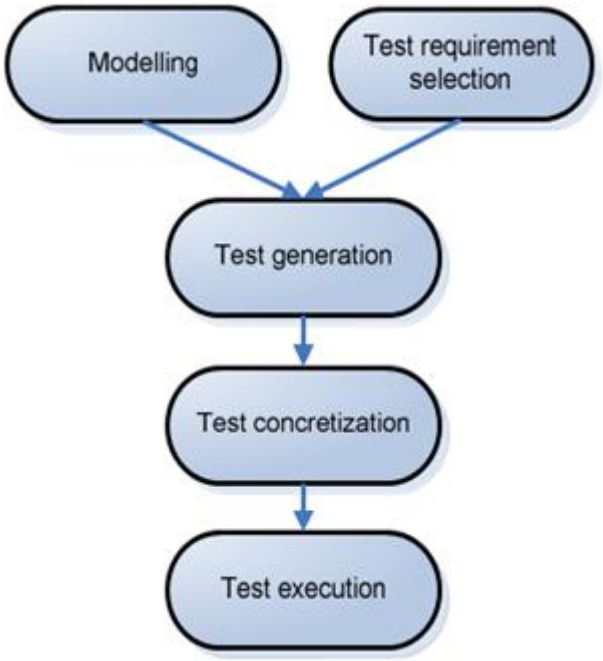

Figure 1. MBT Framework

II. SOFTWARE TESTING EVOLUTION

Software Testing is evolved from manual testing to scripted testing and to Model Based Testing. Figure 2 depicts the software testing evolution. Manual Testing could be a form of software testing, where software testers manually run test suit, without any test automation tools are used for defects detection. Scripted testing could be a form of software testing, where comprehensive test scenarios are developed by test case designers, which testers convert into scripts. Scripted testing rigid control based on innermost knowledge of the software System Under Test and are probable to yield a precise set of testing outcomes. The MBT could be a type of software testing, wherever software testers automatically generate test suit from the models. The models are the predictable performance of the software System Under Test and models are frequently wont to describe the testing stratagem. The model is an idea of the real world situation.

Revised Manuscript Received on December 05, 2019.
***** Correspondence Author
G. Swathi*, Department of Computer Science and Engineering, RNS Institute of Technology, Bengaluru, India.
E-mail: swathigunanchi@gmail.com
Dr. Girijamma H. A., Department of Computer Science and Engineering, RNS Institute of Technology, Bengaluru, India.
E-mail: girijakasal@gmail.com

Figure 2. Software Testing Evolution

III. MODEL BASED TESTING APPROACHES

There are two approaches for Model Based Testing

A. Offline Model Based Testing

Offline / a Priori Model Based Testing generate a finite set of test cases and execute those later. Figure 3 Shows the Offline Model Based Testing. It supports automatic test case generation. Offline Model Based Testing permits automatic test execution in third party test running platforms.

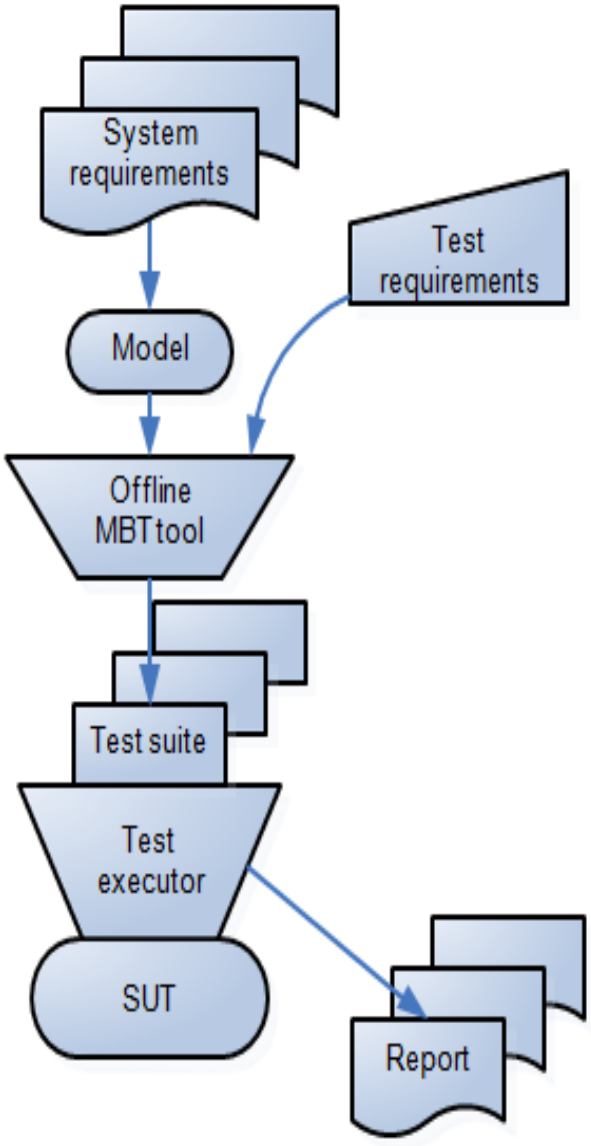

Figure 3. Offline MBT

B. Online Model Based Testing

Online / On-The-Fly Model Based Testing generates test cases during test execution. Figure 4 depicts the Online Model Based Testing. With this approach test suit generation and execution in movement. Here subsequent activity is design when the output receiving. Online Model Based Testing approach is usually used for testing non deterministic systems. It supports infinite test cases running.

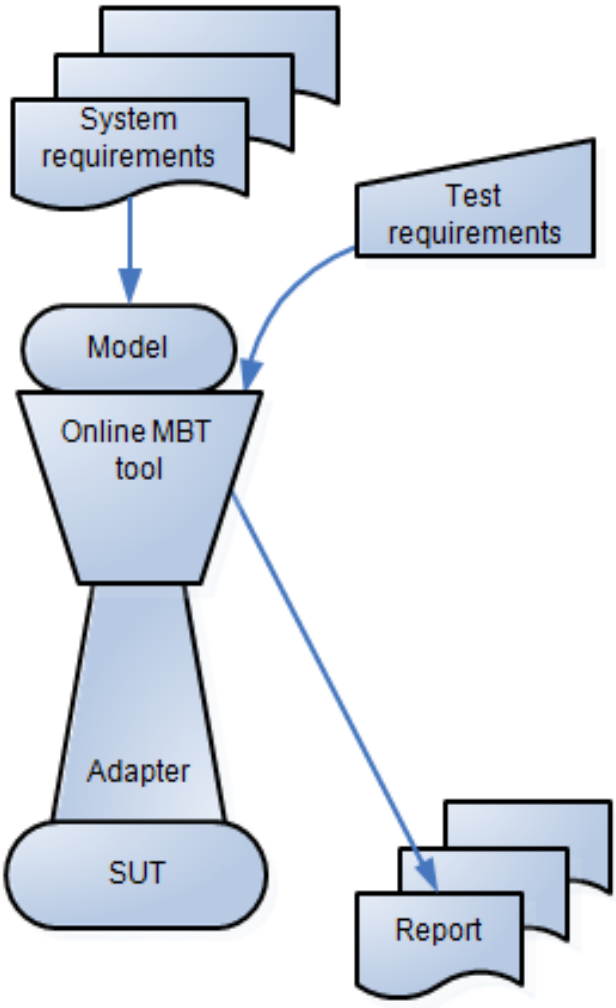

Figure 4. Online MBT

Figure 5 shows the comparison between the Online MBT and Offline MBT.

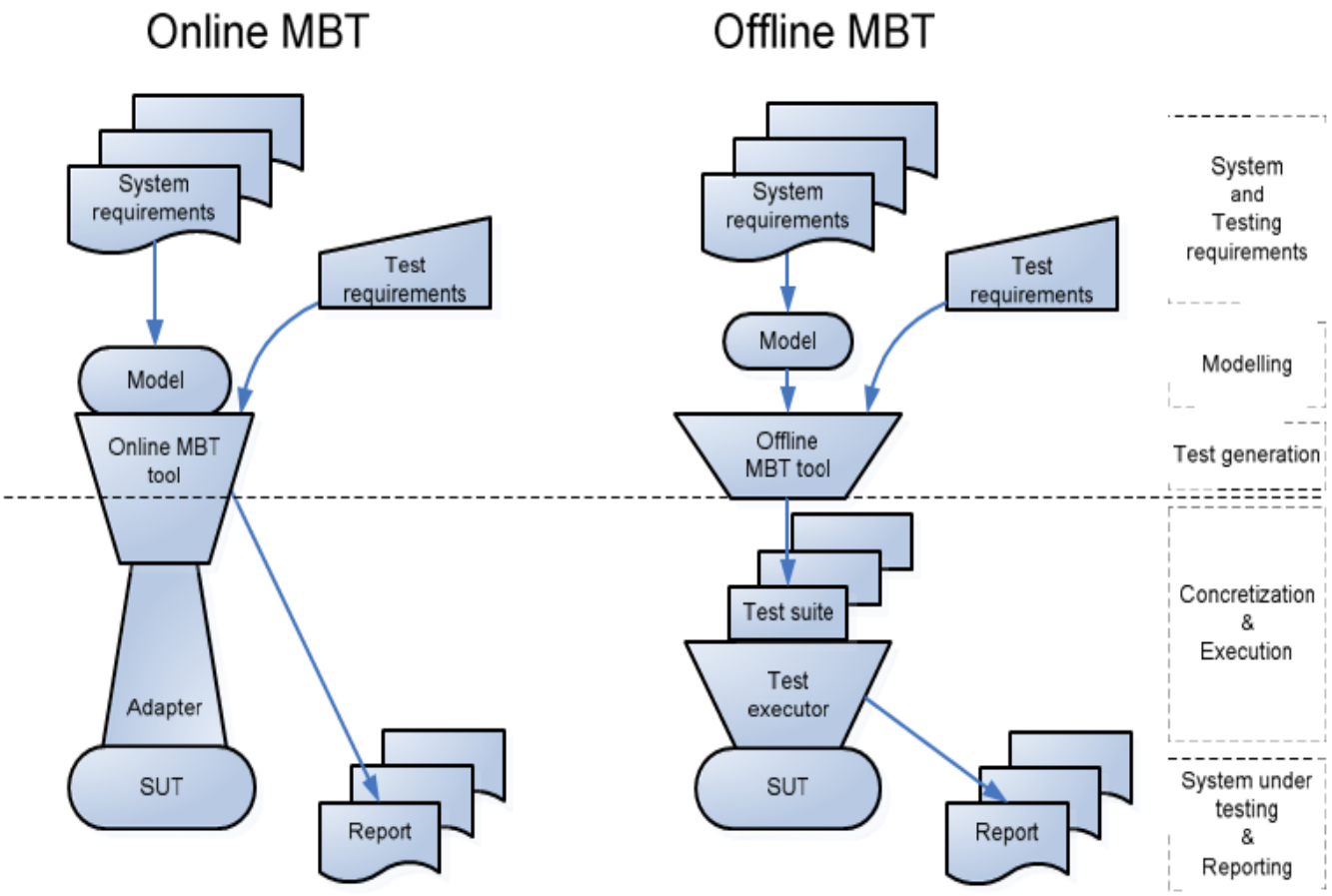

Figure 5. Online MBT and Offline MBT

IV. MODEL BASED TESTING PROCESS

Model Based Testing is an implementation of model based design for producing test cases and running these in contradiction of system under test for testing purpose. This Model Based Testing Process can be usually alienated into 5 activities. That is depicted in Figure 6. MBT process.

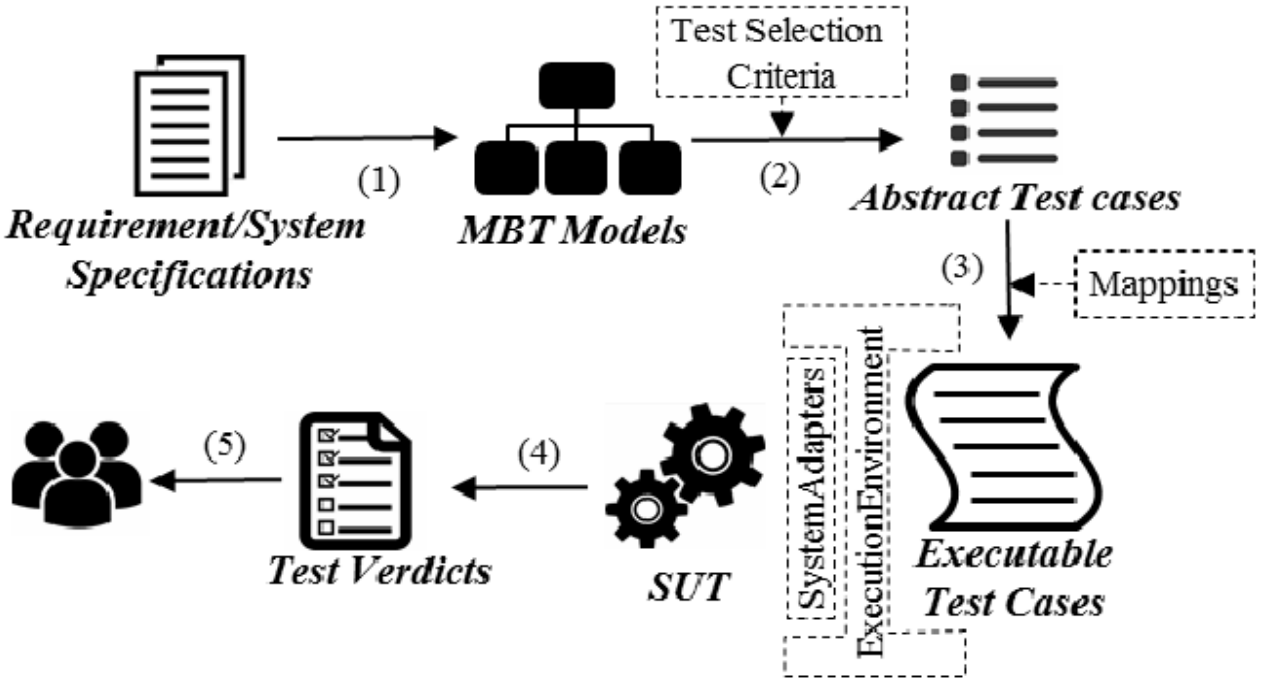

Figure 6. MBT Process

Activity 1: Creation of MBT Models:

The model typically refers to as test model, that signifies the predictable activities of the system under test. In this activity, testers create Model Based Testing models from application specification or requirements. The requirements describe the testing specifications or the features to test of system under test. The created Model Based Testing models typically signify sophisticated abstraction of System Under Test and these are delineated by notations (e.g., Unified Modeling Language and PetriNet) or formal languages. The arrangement of MBT models depend on the features of system which is under testing and also the essential input sequence of MBT tools [14].

Activity 2: The Generation of Test Cases:

This activity 2, once applying the Test Selection Criteria, the conceptual test suit produced automatically from the Model Based Testing models. The Test Selection Criteria direct the test production procedure by representing the fascinating focal point to test, like definite roles of the system which is under testing or the structure of the Model Based Testing model (such as data flow coverage, transition coverage and state coverage). Implementing totally dissimilar criterion to the identical Model Based Testing model can produce different sets of test suit. From the Model Based Testing models an abstract test suits are generated while not any implementation details of system under test [14].

Activity 3: The Test Cases Concretization:

This activity 3, concretize the conceptual test suits from generation of test cases to runnable test suits with the assistance of mapping between the abstractions in Model Based Testing models and details of system's development. Runnable test suit consists the details of low-level implementation, and this may be directly execute against system under test [14].

Activity 4: The Test Cases Execution:

This activity 4, Executes the runnable test cases against system under test, either inside an automatically test case execution domain or manually. The System Adapters are needed to produce channels linking system under test to the test suit Execution Environment, to automatically execute the test suits. While executing the test suits, system under test is stimulated correspondingly by every test case inputs, and also the reaction of system under test like performance information and outputs. These are composed to produce Test Verdicts. The Test Verdict is produced for every Test Case, demonstrating if a test case indecisive or fail or pass. [14].

Activity 5: The Analysis of Results:

This activity 5, The test's outcomes are conveyed at the end as results, to users. The Model Based Testing Process notes suggestions to relate essentials from system requirements specifications to Model Based Testing Models and then to the test suits, that are accustomed to recover attainable faults, for non-satisfactory Test Verdicts [14].

V. BENEFITS OF MODEL BASED TESTING

Model Based Testing benefits are given below:

- Easy to generate Test Cases
- Improves Test coverage
- Reduces in Cost for Testing
- Easy maintenance of Test Suits
- Can execute dissimilar test cases on many no. of systems
- Possible to detect defects at early stage
- Increases defects count
- Can save time
- Improves job satisfaction of testers

VI. RESULTS AND DISCUSSIONS

An implementation of MBT in each organization clearly needs a great quantity of effort and investments. The drawbacks of MBT in Software Engineering are given below:

- More time will be taken to learn
- Essential Skills are needed in tester
- Understanding the model itself is difficult

The placement of MBT into an organization necessitates major quality of investments and effort. This factor approved by almost all analysis on Model Based Testing. The subsequent causes for the fascinating investments and efforts are specified:

Extreme quantities of skill set are essential from the testers:

The organization required to be familiar with the models, that suggest information about diverse varieties of Automata Theory, Formal Languages, Unified Modeling Language, and State Machines. In accumulation, once test automation is going to be used, the knowledge in scripts and tools is crucial for testers [18].

A great preliminary exertion in ters of person-hours is essential:

The kind of the system model has got to be cautiously elected, totally dissimilar components of application must be alienated, so the modelling will be simpler due to the reduced area and therefore the real model has got to be engineered [18].

Model itself has conjointly numerous drawbacks:

The vital one in all those is the outburst of state space needed. The management of the model becomes tedious work and complex, because even a smaller application will hold numerous states [18].

VII. SUITABLE APPLICATIONS OF MODEL BASED TESTING

Before anyone starts to make use of MBT, they need to make certain that this technique is appropriate for their applications or environments.

The building of test model is evidently a huge up front price, however it should be recovered with the reduced evolution expenses once that software system is ready. Surely, if the software application is assessed to own small evolution expenses, after that test modeling prices mustn't be recovered

by the possible system evolution reserves of MBT till abundant later within the software evolution activity. Reduced system evolution expenses could also be foreseen, as an example, if the software system is deliberate to own solitary a brief functioning life time or it's predictable that there'll be mostly limited changes to the software system needed by the environments and its users. Apparently, the software system necessity be suitable for system modelling in a very supporting notation. The software system ought to even be well-thought-out vital enough to warrant the price of MBT. The MBT are improbable to be effectual price, if prime quality isn't essential to the application user [18].

The difficulty of executing ancient testing of concurrent software applications might conjointly operate as the one that drives towards usage of MBT, wherever the combination of the several tests and efficient test suit production offers for the complex software system models, a respectable test coverage [18].

VIII. CONCLUSION

Model Based Testing is a budding and developing methodology for software testing process, that permits software testers to automatically produce software test suits from specific descriptions of a software system's behavior. Automatic test suits generation is one among the most significant benefits of MBT technique. Whenever the software system changes, it's required only to modify the system model to produce new test suits. It is because of the test suits are produced from the model of the software system. This benefit makes Model Based Testing very easy to manage, analyze and evaluate than the ancient automated test suits. Testers are enthusiastic and intelligent to generate Model Based Testing programs will generate versatile, suitable test suits for the price of a non specialized testing language tools. Smart software system tester's cann't stay away from application models. Model Based Testing has developed as a beneficial and effective testing technique for appreciating acceptable test suit coverage of the software systems.

REFERENCES

1. S. Dalal et al. (1999), "Model-based testing in practice", In: ICSE'99, May, pp. 285--294.
2. A. Pretschner (2005), "Model-based testing", In:ICSE, p.722723.
3. El-Far, I. K. & Whittaker, J. A. Model-based Software Testing. In: Marciniak, J. (ed.), Encyclopedia on Software Engineering, Volume 1. New York, USA: John Wiley & Sons Inc, 2001. pp. 825-837. ISBN 0-471-21008-0.
4. G.M. Lima, G.H. Travassos (2005), "A Strategy for Object Oriented Software Integration Testing". In: LATW'2005.
5. Marius Nita, David Notkin "White-Box Approaches for Improved Testing and Analysis of Configurable Software Systems" IEEE 2009.
6. M. Vieira, J. Leduc, B. Hasling, R. Subramanyan and J. Kazmeier (2006), "Automation of GUI testing using a model driven approach", In: AST'06, ACM Press.
7. Blackburn, M., Busser, R. & Nauman, A. Why Model- Based Test Automation is Different and What You Should Know to Get Started. International Conference on Practical Software Quality and Testing, Washington, USA, 2004.
8. J. Boberg, "Early fault detection with model-based testing," in Proceedings of the 7th ACM SIGPLAN workshop on ERLANG, 2008, pp. 9–20.
9. Goutam Kumar Saha "Understanding Software Testing Concepts" ACM 2008.
10. C. Gebizli and H. Sozer, "Improving models for model-based testing based on exploratory testing," in Proceedings of the 8th IEEE International Computer Software and Applications Conference Workshops, 2014, pp. 656–661.
11. Safford, E. Test Automation Framework, State-based and Signal Flow Examples. Twelfth Annual Software Technology Conference, Salt Lake City, USA, 2000.
12. Apfelbaum, Larry. "Model-Based Testing", Proceedings of Software Quality Week 1997.
13. Model-Based Engineering Forum, http://modelbasedengineering.com/
14. Wenbin Li and Franck Le Gall, "A Survey on Model-Based Testing Tools for Test Case Generation" (2017).
15. Kramer, A., Legeard, B.: Model-based testing essentials: guide to the ISTQB certified model-based tester foundation level. John Wiley & Sons Inc, Hoboken, New Jersey (2016).
16. Utting, M., Pretschner, A., & Legeard, B. (2011). A taxonomy of model-based testing approaches. Software Testing, Verification and Reliability, 22(5), 297–312. doi:10.1002/stvr.456
17. Ibrahim K. El-Far and James A. Whittaker "Model-based Software Testing "(2001).
18. Sanjeev Dhawan, Nirmal Kumar and Shiva Saini "Model Based Testing considering steps, levels, tools & standards of software quality" (2011).

AUTHORS PROFILE

G. Swathi, Research Scholar, Assistant Professor, Department of Computer Science and Engineering, RNS Institute of Technology, Bangalore.

Dr. Girijamma H. A., Research Supervisor, Professor, Department of Computer Science and Engineering, RNS Institute of Technology, Bangalore.