



DESIGN OF TEST SELECTION AND PRIORITIZATION SYSTEM FOR MOBILE APPLICATION ENVIRONMENT

Rakesh Gupta, Research Scholar, I.K.G.P. T. U. Kapurthala

Dr. Rajnish Talwar, Principal (CGC-Technical Campus), Jhanjheri(Punjab) INDIA

Dr. Bharat Bhushan, Assoc. Prof. & Head, Deptt. Of Comp. Sc. & Application, Guru Nanak Khalsa College, Yamuna Nagar (Hry.).

Abstract: *The developing significance and stringent quality requirements of the systems of software are expanding interest for effective software testing. Contracting more test engineers or stretching the time of testing is not suitable solutions which are long-term; rather there is a requirement to diminish the required amount of resources. One alluring solution to this type of an issue is test automation i.e., distributing certain tasks of testing to computers. There are incalculable approaches for handling automation of test, and they operate contrastingly in distinctive settings. Creation and designing of test case is very important phase in product. When there is a team hired for working of the development of a safety critical system which involves life risk or health related risks for the customer designing a complete optimized set of test cases in lesser timeframe using existing test case and library of test case is always tough and if completed successfully, can help in achieving the target of Cost, Time and effort reduction in any software project lifecycle.*

Keywords: Test Cases, Test Plan, Software, MES

I. INTRODUCTION

As systems of software turn out to be more and more complex, they are inclined to have increasing number of defects. The expanded complexity of system challenges testers and developers to keep up quality control over the rising artifacts of software and the process of development with test plans which are efficient and effective. While exhaustive software testing can guarantee the deployment of software which is of high quality, exhaustive testing is never practical because of the critical costs of running numerous test cases. Interestingly, if the testing of the software is done nor properly, then failures during the genuine deployment of the software can prompt critical costs included in software fixing, liabilities which are legal and potential and loss of goodwill. What is required is an ideal level

of testing that adjusts the dangers if failures with the expenses caused while testing of software is done to full fill reliability requirements of software.^[1]

Software Testing is a discriminating component of software quality assurance and speaks to a definite survey of specification, design and generation of code. The expanding perceive ability of software as an element of system and the orderly “costs” connected with failure of software and spurring strengths for all around arranged, thorough testing. The general way for testing is represented by the test plan. The main part of the effort of test planning is centred on making the cases of test. A test case is a portrayal of particular collaboration that a tester will have, keeping in mind the end goal for testing a solitary conduct of the software. Black box testing spotlights on the software’s functional requirements. That is, black box testing empowers the software engineer to determine sets of contributing conditions that will completely practice all functional prerequisites for a system. There is a method of black box testing known as equivalence partitioning which partitions the program’s input domain into data classes from which derivation of test cases can be done. A perfect test case having no assistance reveals error classes that may some way or another oblige numerous cases which are to be executed before there is an observation of general error.^[2]

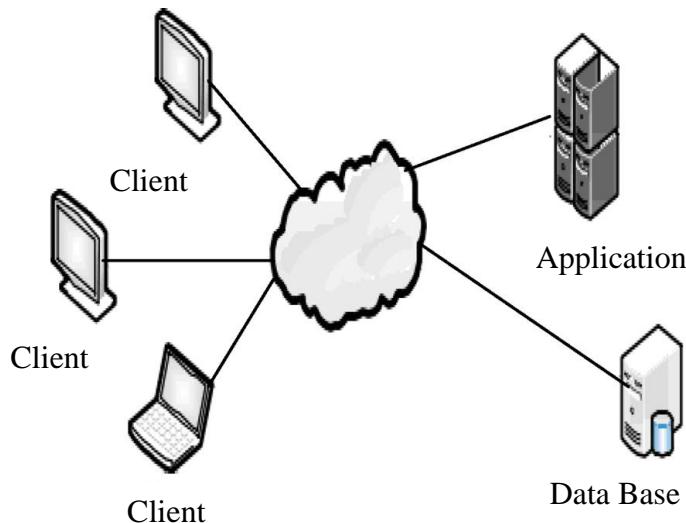


Fig. 1 The system deployment diagram ^[4]

Generating, gathering, examining and devouring test data are critical to corporate business intelligence. Dealing with the making of test plans, automation of test, test data and usage of exceptionally costly resources turns into a process and discipline it could call its own. Automation of these abilities is a key segment of the today procedure of testing. These tests



are frequently complex to do manually and dreadfully draw out to be executed in an environment of manufacturing. This database framework and tools for test wrapped around this procedure is alluded to as framework for automation of test. This framework is the way for improving the efficiency of complete group and provides organizations a competitive benefit by permitting them to deliberately characterize test plans and afterwards rapidly move through the procedure of automating the test plan and creating data of product which is critical. The industry of Automatic test Equipment (ATE) has pushed to build a framework which would hold the sharing of information of test, results and data crosswise over the different platforms of enterprise. Generally, the test engineer adds to a test plan and characterizes the equipment and requirements of software. In view of the test plan, the equipment of test is assembled and group of testing organizes the automation and hence gathers, examines and reports the information when completion of test is done. The process of test for any given organization can incorporate the accompanying stages: (1) Development of test plan, (2) Selection of interface and equipment, (3) Creation of automation plan and development of automation software, (4) Collection of data and its storage, (5) Analysis of data, (6) Assembly of data sheet/test report and also its publication, (7) Test monitoring of supply chain – manufacturing of contract, (8) Utilization of test station, scheduling and reservation, (9) Tracking and calibration of test asset, (10) Manufacturing Execution System (MES).^[3]

II. TEST CASES

A test case is a portrayal of a particular interaction that will be held by tester, keeping in mind the end goal to test a solitary conduct of the software. There is genuine information in the test case that ought to be entered into the software and the expected result that must be generated by the software. A particular test case consists of these sections, generally laid out in a table^[4]:

1. A unique number and name.
2. A prerequisite that this test case is working out.
3. Preconditions that depict the condition of the software prior to the test case.
3. Steps that depict the particular steps that makes up the association.
4. Expected results that portray the normal state of the software after execution of test case is done.



Designing Test Cases: Functional system and acceptance tests are composed by test engineers by utilizing requirements of system and techniques of formal test design such as analysis of boundary value and equivalence partitioning. There come some difficulty in designing good test cases and it is one of the primary skills which must be possessed by expert test engineer. There are likewise approaches for automating the process of test design. Expected results for tests can once in a while be produced consequently utilizing which are known as oracles, external trusted entities which can be questioned for expected results. Yet an oracle is some current framework, they can likewise be made only for purposes of testing. The tools can likewise create cases of test which are dependent upon code or interfaces of software. In testing which is model based, the modelling of system is done in such a subtle element that the derivation of test cases can be done automatically from the model. [5]

III. RELATED WORK

This section reviews the existing work on the testing of software and test cases in mobile environment.

In Paper [6], the importance of keeping high level of abstraction within automated test and its characterization is discussed. In order to keep an automated test suit running even though an extensive development process is in progress is a crucial task for efficient software development. It talks about how this abstraction should be constructed and used. The abstraction designs the same for both input and output. Also, when talking about abstraction, the main design of the test suits is important. The other tests, that do not concern the added attribute will run and pass. That is, if tests are constructed in such a way that every single test focus on just validating the output relevant for this test, and not all tests validate all the output every time, then the abstraction would increase and the maintainability effort would be lower.

In Reference [7], authors discussed about the connected portable applications development. It is a complex task due to device diversity. Therefore, the approaches which are independent of device are aimed at concealing the differences among the distinct mobile devices in the market. This work introduces DIMAG, a framework for software for generation of mobile applications which are connected for platforms of multiple software, following a declarative approach. DIMAG provides transparent data and state



synchronization between the server and the client side applications. The proposed platform has been designed making use of existing standards, extending them when a required functionality is not provided.

Reference^[8] explained an approach for mobile applications which are heterogeneous. To adapt to failures and changes which are unpredictable, additionally with the different settings offered by the plenty of accessible devices, mobile applications should be as versatile as could be expected under the circumstances. On the other hand, mainstream strategies of adaptation are normally characterized significantly and oblige complex control procedures emphatically interlaced with the logic of application, respecting applications that are not easy to build, keep up, and evolve. This issue is solved by proposing an approach which is declarative to deal with adaptive mobile applications which are also heterogeneous. The benefits of this strategy are demonstrated through an example roused by a mobile application which is currently distributed worldwide, whereas the implementation of the proposed methodology has been accepted through an arrangement of experiments and simulations aimed at representing its performance.

In Paper^[9], authors talked about mobile software applications. The abundance of mobile software applications (apps) has created a security challenge. These apps are widely available across all platforms for little to no cost and are often created by small companies and less-experienced programmers. The lack of development standards and best practices exposes the mobile device to potential attacks. This article explores not only the practices that should be adopted by developers of all apps, but also those practices the enterprise user should demand of any app that resides on a mobile device that is employed for both business and private uses.

In Reference^[10], authors presented a mobile application which is context appropriate for sustainable weight loss. A mobile application is designed win this work that influences three methodologies from theories of change of behaviour which are also proven (expanding awareness of the points of dieting, cultivation self-efficacy and inspiration, and affecting attitudes of dieters). We adjust the application to the nearby context of the society of a middle easterners by leading an ease of an experiment of usability testing with potential clients of the application. The principles of localization are applied additionally to infer an application which is appropriate. Beyond the connected usage of the application, the paper



adds to at present rare assortment of literature on mobile development which is Arabic based.

IV. MOTIVATION

Method of designing new test cases and choosing existing test cases directly affects profitability, efficiency and effectiveness of any product or system. Specially in the field of safety critical embedded systems and consumer electronics, having a mechanism to create automatically test cases which are new from given parameters and effectively choosing the appropriate test cases and results of test from existing test case library can help in achieving overall product process improvement, Quality and safety achievement and reduction in cost, efforts and times. The main problems which are solved in this research work are as follows:

1. When Independent Verification & Validation tries to optimize testing “How test cases ought to be selected and how test priority should be managed?”
2. Finding an effective design for an automated and intelligent system complete test case management and creation of automated test case which reduces the complexity of managing and creating accurate test data.
3. Effectively providing an interface for this system so that it can be plugged easily with different critical systems and consumer electronics goods.
4. How testing should be focused i.e. which parts of the products should be tested more thoroughly?
5. To optimize testing, the most fundamental question that has to be answered is from the test suite, which of the test cases should be executed?

The major objectives of this research work are given below:

1. To study and compare the existing testing system for mobile application environment.
2. To propose an efficient test selection and prioritization system for mobile application environment.
3. To design appropriate scenarios for testing of proposed systems.
4. To verify and validate proposed approaches using appropriate test.

V. PROPOSED WORK

The approach is based on automated generation/creation of test cases from specific inputs and management of test plan based on existing test cases and plans on similar product line.



Proposed system will analyse the key parameters of a test plan, data of test result, test case, etc. This analysed data will be stored in a repository and a user friendly extraction software utility will be proposed which will make new test cases of similar product line in a short manner. Proposed system will also use an intelligent approach to make test cases which are new based upon previous test experience and gray coverage areas.

The main parameters of standard test case are given below:

1. Type of test case
2. Requirement Id
3. Test case type
4. Feature/Function/Category
5. Precondition
6. Inputs
7. Input Criteria
8. Procedures
9. Expected behavior result

In this research work, we have proposed a complete system for advanced test plan creation, management and extraction tool (STP Star) for High End embedded systems and safety critical systems. The proposed solution will cover and extract up to 18 different types of STP and will feature a color touch-screen, Avionics Bus systems, DO-178B compliance, MCDC, mobile application test , Learning, Activity control, Profiles etc.

The thesis enlighten an approach of creating new test plan, generation of test cases from existing test cases can be prepared which will use intelligent approach to generate efficient test sets from existing test plans on the basis of Entry criteria, exit criteria and some other selected parameters. The proposed design and algorithm has been implemented in such a manner that it can be used for:-

1. Creating a set of test cases for similarly existing products/slightly variant.
2. Creation of test cases for embedded products like Safety critical systems, Mobile phone, remote control and multiple consumer electronics goods.
3. This algorithm takes care of changes in the test plan template and automatically configures the test database according to a new format.

4. It can be easily clubbed with a plugin information system to have a complete automation solution for product testing.

The basic software architecture approach for admin and user levels of the test plan creation, extraction and management system are shown in given figures.

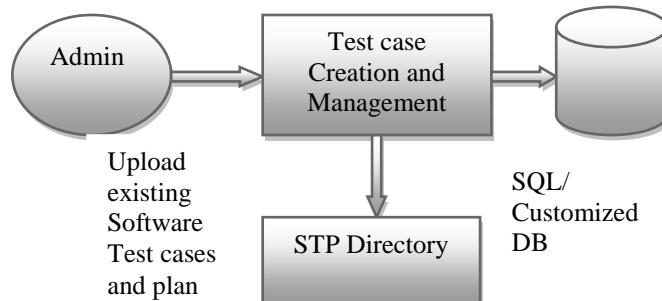


Figure 2

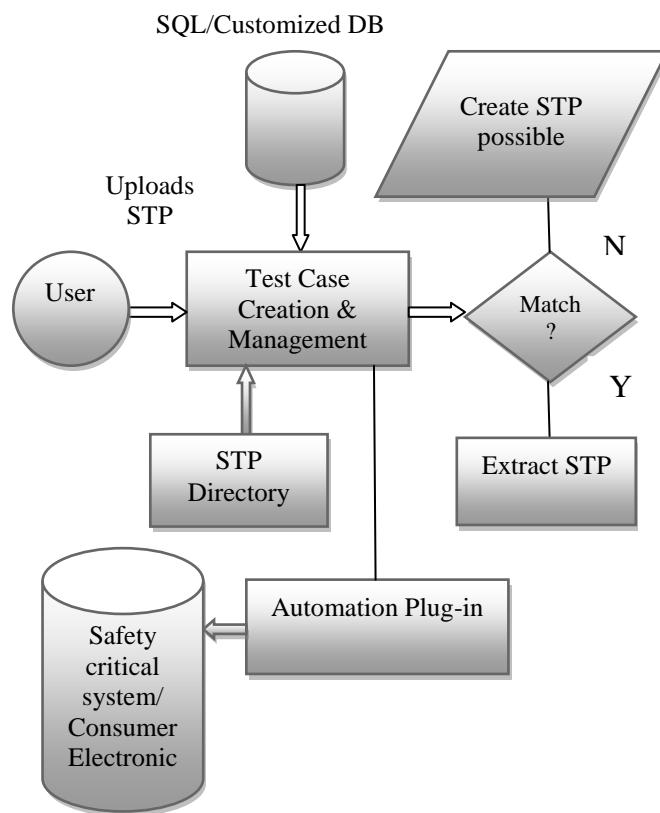


Figure 3

VI. CONCLUSION

In this research work, we will demonstrate that an effective testing system and environment which has the capability for creating new test cases for complex products in an automated manner without introducing manual error and use the existing intelligent test base



integrated with an automation system can achieve a faster test cycle which is more effective as compared to a generic manual or pure automation cycle. This research work demonstrates that an effective test environment with generation of test case, test automation and test intelligence capability can be used for product development. This will ensure a defect free safety critical or consumer electronics goods as well as a happy customer.

ACKNOWLEDGMENT

The paper has been composed with the kind assistance, guidance and support of my guides who have helped me in this work. I would like to thank all the people whose encouragement and support has made the fulfilment of this work conceivable.

REFERENCES

- [1] Kaushal Chari, Alan Hevner, "System test Planning of Software: An Optimization Approach", IEEE Transactions on Software Engineering, Vol. 32, No. 7, July 2006.
- [2] Hao Wu, "An Optimized Design for the Test Case of Web Application", IEEE Symposium on Electrical & Electronics Engineering (EEESYM), IEEE, 2012.
- [3] Michael Weir, Ross Kulak, Ankur Aggarwal, "Service Oriented Architecture for Agile Automated Testing Environment", IEEE, 2013.
- [4] Guixia Yuan, "Study of Implementation of Software Test Management System based on Web", IEEE, 2011.
- [5] Pekka Laukkanen "Data-Driven and Keyword-Driven Test Automation Frameworks", Helsinki University of Technology, Department of Computer Science and Engineering Software Business and Engineering Institute Feb, 2006.
- [6] Andreas Lundgren, "Abstraction Levels of Automated Test Scripts", Department of Computer Science at Faculty of Engineering Lund, Sweden and Cinnober Financial Technology Stockholm, Sweden Mars, 2008.
- [7] Patricia Miravet, Francisco Ortin, Ignacio Marin, Abel Rionda, "Using standards to build the DIMAG connected mobile applications framework", Elsevier, 2013.
- [8] Gianpaolo Cugolaa, Carlo Ghezzia, Leandro Sales Pinto, Giordano Tamburrellib, "Self Motion: A declarative approach for adaptive service-oriented mobile applications", Inc. Elsevier, 2013.



- [9] Stephen M. Dye, Karen Scarfone, "A Standard for developing secure mobile applications", Elsevier B.V., 2013.
- [10] Selma Limam Mansar, Shashank Jariwala, Maahd Shahzad, Aysha Anggraini, Nawal Behih, Amna AlZeyara, "A Usability Testing Experiment for a Localized Weight Loss Mobile Application", Elsevier, 2012.
- [11] M. Egmont-Peterson, D. de Ridder, H. Handels, "Image processing with neural networks", Pattern Recognition 35 (2002), pp: 2279-2301, Elsevier, 2002.
- [12] Yusuf Abu Sadah, Nijad Al-Najdawi, Sara Tedmori, "Exploiting Hybrid Methods for Enhancing Digital X-Ray Images", The International Arab Journal of Information Technology, Vol. 10, No. 1, January 2013.
- [13] Howard Lee, Yi-Ping Phoebe Chen, "Image based computer aided diagnosis system for cancer detection", 0957-4174/ 2015, 5356-5365, Elsevier, 2015.
- [14] Francesco Ciompi, Colin Jacobs, Ernst Th. Scholten, Mathilde M. W. Wille, Pim A. De Jong, Mathias Prokop, Bram van Ginneken, "Bag-of-frequencies: A Descriptor of Pulmonary Nodules in Computed Tomography Images", IEEE Transactions on Medical Imaging, Vol. 34, No. 4, April 2015.
- [15] Dr. K. Meenakshi Sundaram, D. Sasikala, P. Aarthi Rani, "A Study on Preprocessing A Mammogram Image Using Adaptive Median Filter", International Journal of Innovative Research in Science, Engineering & Technology, Vol. 3, Issue 3, March 2014.