JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AN INTELLIGENT ASPECT ORIENTED FRAMEWORK FOR TESTING MOBILE APPLICATIONS

[1]Prathap D L, [2]Shruthi N, [3]Shivi Dixit, [4]Manasa K, [5]Pooja P S

[1]Assistant Professor, [2] Assistant Professor, [3] Assistant Professor, [4]Assistant Professor, [5]Assistant Professor
Mysuru, Karnataka.

Abstract: Software Testing process is composed of many key challenges like reducing cost of testing, reducing test cycles, increase in test coverage and improving quality and efficiency. Many companies consider testing automation as the crucial step in establishing a mature QA (Quality Assurance) program and certainly have lot of value if it is effectively leveraged. Test framework is the basic infrastructure required for running software test scripts and consolidating test runs. Small to medium sized companies usually adopt generic test automation frameworks and design their logic to accommodate the company specific business needs. There are different Framework architectures available today but they do not cater to all the needs. In this paper we have analyzed the problems faced during the developing of test framework. The main purpose of this paper is to provide a clear understanding of what is required in starting the development of any mobile application testing framework, defining test scenarios, and automating them and also provides ABF (Aspect based framework) which fits the present and future business needs and ever-changing requirements.

***Keywords*—Testing process, Test Framework, Mobile Application under Test (AUT), Efficiency Quality (EQ) Quality Assurance, Aspect Based Framework (ABF), Test scenario.**

I. INTRODUCTION

Test automation refers to the development and usage of tools to determine the success or failure of pre-specified test cases, against the Application under Test (AUT), without human input. The primary objective of test automation is to reduce repetitive manual tasks. Mobile Application testing is a continuing challenge because of dynamically changing requirements and the need for increased test coverage. Test engineers test products with respect to functional and non- functional requirements. The most important issues of the test framework are its simplicity, maintainability, flexibility, generality, reusability, performance and scalability in a specific area. Mobile Application Software that constantly updates and improves makes it especially difficult for these points [1].To create and run test cases that verify whether a product can run as expected in its operating environment, the engineer should be able to identify the requirements. It is necessary to maintain a centralized test automation framework over geographical boundaries. Automation framework is an infrastructure containing a set of utilities that helps to develop test automation for any application, and helps to scale and adapt to the changes in the application with minimal effort. A test framework design is a set of assumptions, concepts and practices that provide support for automated software testing [12]. These frameworks enable test engineers to build efficient test cases and to execute them as well, helping to achieve required quality and to improve operational efficiency. Nowadays because of being restricted by progress, budget, man power, resources and so on, software testing occupies a large portion in software life cycle. In view of decreasing testing resources and improving testing efficiency [2] Complexities of today's software require robust strategies for achieving enhanced speed of test execution at a lower cost. Test Automation enables organizations to meet this objective and also yields high return on investment. An automated testis faster, because it needs no human intervention [13]. As computer systems are permeating our society in daily life and are performing an increasing number of critical tasks, research in software testing and analysis has become of paramountimportance. Although we are currently not able to prove program correctness for real-world applications, rigorous software development processes in combination with testing provides us with confidence in the quality of software. Software testing and analysis, however, is a very involved task. As the size and complexity of software continue to grow, manual testing becomesvery tedious. Automation of software testing and tool support for testing, therefore, has been emerging as a key technology to quality assurance of today's software industry. Measure of qualityis the inverse of the defect Density used in many previous quality studies [16]. As research in software testing and analysis has become increasingly active, there is also a growing trend towardscombining formal methods and informal techniques for program Verification [3]. Whichever testing method we choose, a framework would be required to implement the strategies, and building a test framework is facing the following problem [1].

Mobile phone system is not stable. Different mobilephone manufacturers have different opinions in the platform selection and R&D, resulting in the marketdiversity of mobile platforms. Even the same phone vendor may cover multiple platforms to create their different positioning of products. In addition, platform is alsoconstantly upgrading. Mobile phone system as a mobile

application software running platform directly restricted the means of tests implementation, thereby affecting the overall framework design.

- Mobile application software updates frequently. Changes in mobile phone system are one of the reasons for application updates. But lacking of a universal and unified architecture for mobile application software development is the major reason, and more important this type of update leads to structure, data flow differences between versions and moreover the substantial adjustments of testing methods, test tools and test cases which can often be on the basis of them.
- For different applications to vary the demand for testing. Different applications have their own unique business flow and data flow, this difference inquires asophisticated testing framework [1][4][5][7][8].

BACKGROUND

Few automation frameworks do not scale to meet the needs ofa test organization over a period of time. Automation frameworks tend to become a throw-away prototype rather than a sustaining infrastructure. This section gives an overview about the general issues and problems faced during design of an automation framework [12]. Because of different characteristics of AUT there are restrictions in the process of Mobile application testing. That means we should development different Testing environment for different AUT or even for different testing requirements for the same AUT, which cause the specialization of testing environment. The specialization of testing environment leads some problems [9] the design of testing environment lacks the consideration of reusability. Every testing environment will not be used after the testing, which leads lots of wasted work. Special testing environments usually lack sufficient testing because of short using period. It's difficult to ensure the correctness and efficiency of testing because of unreliable testing environment [7][9].At present scenario, knowledge management in the field of software testing is researched infrequently, knowledge management has been just executed in testing [17]. Following are the major issues and bottle-necks faced during the design of test Framework [12].

- Hard testing work for testers because of dynamically changing user screens in mobile applications.
- Deviation in design and scope: Most test automation frameworks which become non functional or obsolete are due to deviation in design or scope as part of the design strategy. Focus is usually on current requirements, and not on long term requirements. No fore-sight on use of frameworks across different business units and teams.
- Simplicity, maintainability reliability, flexibility: Frameworks fail to meet the criteria of Simplicity, Maintainability, Scalability and Flexibility due to reasons like Complexity in the framework in terms of Languages used, Technologies used, Framework design, Hard-coding done in framework. Less modular and re- usable code. Framework built around single technology or being application dependent Maintainability problemsdue to huge amount of code per test case. Unable to accommodate different methods of handling data. Improper testing and qualification of framework anddocumentation. Frameworks scale up the to meetcomplex needs, but are not flexible enough to meet simple needs – this is a major reason for the development of temporary automations and a variety of small tools which add on to the maintainability problems.
- Performance and scalability: Framework design focuses on functional aspects, rather than performance and scalability. Frameworkperformance is critical and it determines how efficient framework is, while running automated testsin a distributed environment. These are not visible during the initial design, but when the framework scales up and is used across teams in parallel, performance issues come to fore
- Obsolescence arising from better and newertechnologies: The technologies are fast evolving and getting better day by day. The technologies used in a framework get obsolete by the time automation starts for the next product.
- Vertical efforts involved and time constraints: Frameworks require huge effort and time in Setup and install. Learning curve and ramp up on the framework. Developing new test cases due to framework details. Adapting to new requirements and needs. Scaling up andscaling down.
- Tools /Technologies/Language: This is one of the most common problems faced. This is caused mainly due to People's personal choices, comfort zone andtechnology/language of strength. Lack of expertise and domain knowledge [12].
- Maintaining heterogeneous and multiple frameworks: This problem is commonly faced by higher management. Within an organization, variousframeworks would be developed based on the functional needs of the business units. It becomes a nightmare to manage all the existent heterogeneous frameworks. This situation may arise due to differentfunctional needs like, people needing frameworks to test: CLI, Simple web-based GUI, Advanced JavaFX based GUI, Applications using proprietarytechnologies, and etc. People's tendencies to create something challenging and complex and not to re-use things and re-invent things in their areas of strength [1][12].

Some popular Software test techniques in the mobileapplication are [1].

- Record/Playback Technique: Recording the events occurred in GUI operation through user's mouse and keyboard input and by recycling recorded events. But if any change to application occurred, all the influencedtest cases should be re-framed.
- Capture/Playback Technique: User uses mobile application software to test by setting an example, andthen writing scenario
- Particulars-Based Test Technique: The particulars are necessary to be explicitly described, to be perfect and to be summarized and designed. To get these particulars makes constant efforts.
- Beta Test Technique: Beta test is the most popular test method. It is used very much as a method carrying out test to common users by merchandising the software of beta version. But it is difficult to get common users to educate and understand which will affect the test effect [4][5][6][7][8].

PRESENT WORK

Proposed framework for mobile application testing is shownbelow.

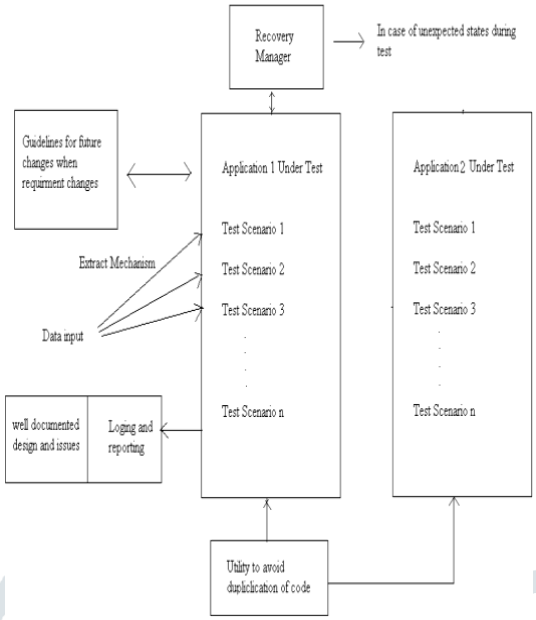

Fig. 1 ABF for mobile application testing

ABF Consists a set of tools used for mobile application testing,this makes the tester's life simple to run the test, In order to liberate the testers from the hard testing work it is important to improve the automatic and intelligent level of the testing execution [14]. With this design the overall effort spent to write a single test case is minimal.

Features:

- Tester can run two kinds of tests namely: functionalexecution and non-functional (performance and load, scalability etc). This can be used to compareperformance across various versions.
- Test scripts and libraries are written to run onJavaFX application.
- It is possible to integrate other frameworks withABF.
- Test scripts and libraries are to be checked into a SCM, so that users can make use of each other's tests, libraries and tools.
- Supports the execution of many test scripts in parallel, including having test scripts run on many remote machines, including message passing among tests.
- The framework can auto-generate test cases based on different versions. These auto-generated test cases act as input and are re-usable.
- Configuration parameters are input through a simple JavaFX GUI.

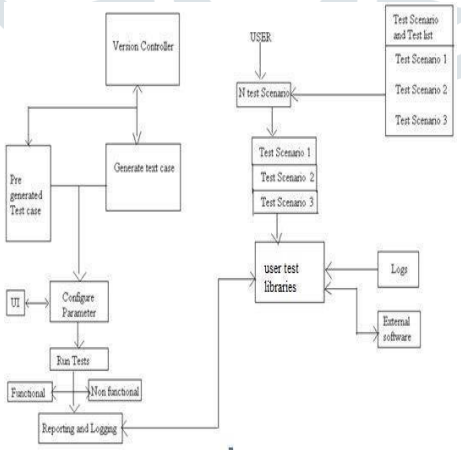

Fig. 1 ABF test execution process

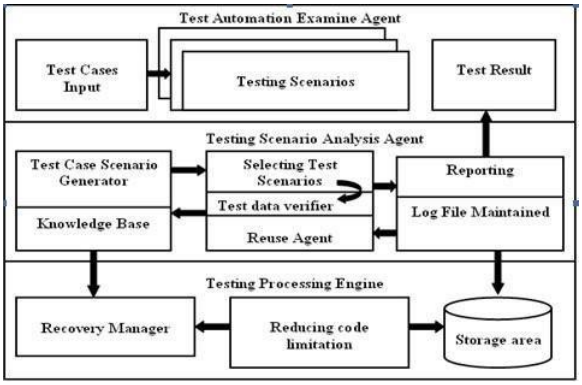

Fig. 3 Proposed Test Architecture for mobile application

The bottom layer (1) Testing Processing Engine consists of Testcase input, Test scenarios component. These will serve as a basis to define specific bridging tasks and to develop appropriate Testing scenarios. The second layer (2) deals with the use of Test Scenario Analysis Agent which verifies the test data that are selected form test scenarios. Simultaneously reporting and log fileis maintained for reusability. Top layer aims at providing reducing code limitation, storage area and recovery manager.

TEST AUTOMATION EXAMINE AGENT

Test automation Examine Agent is the used control the execution of tests, the comparison of actual outcomes to predicted outcomes, the setting up of test preconditions, and other test control and test reporting functions. It automatically selects the test scenarios based on the priorities, and this is sent for testing scenario analysis agent further verification and testing the cases. Concurrently reporting and log file is maintained for future usage.

TEST CASE SCENARIO ANALYSIS AGENT

It is possible to automatically generate checking code for many preconditions as well as for many clauses in post conditions. One possible approach to solving this problem is to automatically generate everything that is appropriate, and allow a human to provide the code for those checks that cannot be automated. Knowledge base is used as an expert in taking the decisions for proper selection of test scenarios and to reuse the existing test case so that it reduces the testing effort. Experience with the prototype wrapper generator indicates it is a simple process to separate the human- contributed checks from all of the other infrastructure code necessary to support.

TESTING PROCESSING ENGINE

Recovery mechanisms in case applications encounter unexpected states during a particular test/scenario. It provides the common utilities which will improve usability to avoid duplication of code. Set of rules on how to develop automation with minimal impact in case of future changes to the application, Logging and reporting mechanisms Logging and reporting mechanisms helps to find out memory leaks and gets user friendly readable reports for specified intervals.

STRATEGY TO BE FOLLOWED WHILE DEVELOPING THE MOBILE APPLICATION FRAMEWORK

This section gives an overview about the strategy to be followedduring the design of a Mobile application testing framework.

- GUI should Simple for configuring parameters and test execution.
- During Regression testing, when tests must be run repeatedly for every change in the application, it is advantageous to have as small a set of test cases as possible [15].
- Present business needs Future Business goals and ROI - Return on Investment are the important aspects to be taken care which decides the success of theframework.
- In order to reduce rework, keep in mind that Frameworks should accommodate needs of multipleentities like teams, products, business and units otherwise it leads to a failure of whole work.
- Regression testing should be given more importance as significant amount of test cases would come under the regression category over a period of time [1].
- Framework development should be treated as an independent project. It should not be considered as something that people can do when they have a little extratime, or between projects [12].
- All aspects of the framework should be well documentedfor future use.

We assume a sample size of N test cases per test cycle. Benefit of using test automation framework for each test cycle is given by

$$I_S = (I_M + I_{NP}) - (I_A + I_{DF})$$ Where,

I_S = Instance saved due to test automation framework

I_M = Instance taken for manual testing

I_A = Instance taken for automated testing

I_{DF} = Instance taken for developing framework (once forDifferent releases of the product)

I_{NP} = Non productivity instance associated with rework

EXPENDITURE OF DEVELOPING TEST AUTOMATION FRAMEWORK(ONCE INVESTMENT)

Expenditure is given by: $\mathbf{E_A = E_{HS} + E_{DM} + E_T}$

E_{SH} = Expenditure of software and hardware

E_{DM}= Expenditure of developing and maintainingAutomation script

E_T = Expenditure of training staff on automationTools

CONCLUSIONS

Framework is feasible because that allows testing development environment maintains high efficiency with good reusability and extensibility. By using this new framework, we can significantly reduce the number of test cycles and test effort required to detect and remove a certain number of software defects and also in the present work It is very clear that an Test framework plays a vital role for the testing success of an organization. So it is critically important to have in place or build an effective Test framework. Bottle necks in framework design include lack of good characteristics, lack of fore-sightedness, lack of insight and people's influence. So it is must to keep things like personal choices, short visions and temporary solutions aside and head straight towards developing a flexible, scalable, re-usable, modular, maintainable, extensible and simple Test framework which can sustain over time and accommodate volatile requirements. To overcome these potential bottle-necks frameworks should use things like Test automation Examine Agent, Test Case Scenario Analysis Agent and Test processing Engine which helps in improving the quality and efficiency of mobile applications.

REFERENCES

[1] Zhi-fang LIU, Bin LIU, Xiao-peng GAO SOA Based MobileApplication Software Test Framework, 978-1- 4244-4905- 7/09, IEEE 2009.

[2] Ping XU, Yangling WANG, Zhong-nan SHEN Application of Software Testability Measurement Model SPM to Software Testing, 978-1-4244-4905-7/09, IEEE 2009.

[3] Zhi Quan Zhou "Automated Software Testing and Analysis: Techniques, Practices and Tools" Proceedings of the 40th Annual Hawaii International Conference on System Sciences(HICSS'07).

[4] Y. Guo, H. Deng, and S.B. Yi, "Service oriented grid softwaretesting environment," Journal of Software, 2006, Vol.17, No.11, pp.2335-2340.

[5] F.Q. Yang, "Thinking on the development of software engineering technology," Journal of Software, 2005, Vol.16, No.1, pp.1−7.

[6] H. Cervanted, and R.S. Hall, "Technical concepts of service orientation. In": Zoran S, Ajantha D, eds. Service- Oriented Software System Engineering: Challenges and Practices. Idea Group Publishing, 2005, 1−47.

[7] Z.B. Chen, J. Wang, W. Dong, and Z.C. Qi, "An interface model for service-oriented software architecture," Journal ofSoftware, 2006, Vol.17, No.6, pp.1459-1469.S.M. Hwang, and H.C. Chae, "Design & implementation of mobile GUI testing tool," International Conference on Convergence and Hybrid Information Technology, 2008, 704-707.

[8] Yongfeng YIN, Bin LIU, Guoliang ZHANG, "On Framework OrientedEmbedded Software Testing Development Environment" 978-1-4244-4905-7/09, IEEE 2009.

[9] H.B Qian, W. Zhao, D. P. Cheng, "Research of Static Detection Technology of Software Testability", Journal of Computer Applications, 2004, Vol. 1, NO.24, pp. 16- 19.

[10] Wu Caihua Zhu Xiaodong Liu Juntao, "Integral Framework of Software Testing Platform for General Support Equipment", The Eighth International Conference on Electronic Measurement and Instruments ICEMI'2007.

[11] Vishal Kulkarni, Yatin Manerker, "Best practices for designing automation framework", proceedings of International Software Testing Conference 2009.

[12] James Bach "Test Automation Snake Oil" published in Windows Tech Journal (10/96) and the proceedings of the 14th International Conference and Exposition on TestingComputer Software, 1999.

[13] Lie Xu, Baowen Xu "A frame work for web application testing", proceedings of 2004 international conference on cyber worlds.

[14] R. P. Mahapatra, and Jitendra Singh "Improving theEffectiveness of Software Testing through Test Case Reduction" World Academy of Science, Engineering and Technology 37 2008.

[15] Ning Nan and Donald E. Harter "Impact of Budget and Schedule Pressure on Software Development Cycle Time andEffort" IEEE Transactions on Software Engineering, Vol.35, No.5, Sep/Oct2009.

[16] R. Brent Gallupe, "Knowledge Management Systems:Surveying the Landscape" [M], Queen's University at Kingston, October2000.