

Testing Complexity in Component Based Software Engineering

Jogannagari Malla Reddy, S.V.A.V. Prasad, Kothuri Parashu Ramulu,

Abstract: *The Component Based software development which reduces the time, budget and efforts compare to the traditional software development. The components are reusable, developed by third parties and available at Commercial-Of-The Self (COTS). Most recent technologies such embedded systems, mobile phones, control systems and other paradigms are developed with this concept. Software researchers should pay more attention on testing of Component Based technology. The software testing is the most commonly used technique for validating the product quality. It is something different from the traditional software development, due to the invisible source code and design make it very difficult to trace the faults. Software testing strategy is road map integrates test case design methods into well planned series of steps for successful construction of software. The selection right strategy at the right time will make the software product as robust. This paper highlights the various software testing challenges, difficulties, strategies and techniques in perspective of Component Based Software Development*

Index Terms: *Modularity, Reusability, Component Based Software Development, Black box testing, White box testing, Regression Testing.*

I. INTRODUCTION

Computer Based Software Engineering motivates the development of software systems with the existing software components which are reusable entities. This evaluation highlights the realization by the software customization and replacement of components. The idea component based construction envisioned by McIlroy more than four decades implemented in other engineering Plot forms. From the last few decades the idea expanded with characteristics such increased efficiency, effectiveness, improved performance, reduced development costs, and time with more quality standards i.e. maintainability, portability on the other side of the development[1].

Component Based Software Engineering includes the various disciplines and concepts of traditional Software Engineering. Many of products have been developed and implemented successfully in the real time environment such

Revised Manuscript Received on September 2, 2019.

Dr. Jogannagari Malla Reddy, Head, Department of Computer Science & Engineering, Indur Institute of Engineering & Technology, Siddipet, Telangana State, India.

Dr. S.V.A.V. Prasad, Dean(R&D), Lingaya University, Faridabad, India.
Kothuri Parashu Ramulu, Rayalaseema University, Kurnool, Andhara Pradesh, India.

as office automation systems, distributed and web based applications. Recent component based technology using disciplined naming scheme such as CORBA, Microsoft DCOM and EJB are typical examples[2]. Microsoft introduced Component Object Model in the early of 1990 as a popular software development paradigm

Component based software engineering is to develop the large scale software systems by integrating software components within the time, budget with less efforts. The aim of the technology is to develop the high quality software using the minimal testing efforts, integration of reliable software components which are chosen from the Commercial-Of-The-Self (COTS)[8]. These are to be reused across the various software products and product families in different environments and must be tested adequately.

Software is becoming increasing complex and there is a need of systematic testing strategies. As per the NIST survey reports, it is found that the annual cost of a software error due to inadequate software testing infrastructure is up to 60 billion dollars and improving the testing infrastructure can save at least one-third of the cost incurred by software errors, or over 20 billion dollars. The testing verifies the software products system against to its specifications

This paper highlights the various technical aspects of component based software systems in innovative direction. The Section 2 explains the taxonomy of component based software Testing. The Section 3 states overview of the component testing spectrum, its strategies. The discussion about various software testing techniques are given in the Section 4 and finally concluded with discussion in section 5.

II. THE TAXONOMY OF COMPONENT BASED SOFTWARE TESTING

The software testing is core area in the software development with many technical aspects. Software testing consumes 40-50% of efforts of development and occupies the significant part in the software engineering. It examines the quality and its maturity of the software..

Software industry demands the products within the time and budget[4]. These are some of the reasons to motive the component based software Engineering to save time and cost by using already build components which are pretested. The components and component based products will be widely used by non programmers for building their products. Automatic component update over the Internet will be a standard means of application upgration which is already in many applications[1].

The component testing has own identity in assessing the software quality In CBSE the components have the following basic properties.

Identity: Each component in COTS has individual and quality assured with unique feature in its effectiveness at the deployment environment. Without this feature large scale of component reuse is impractical and meaningless.

Modularity and Encapsulation : Components are modular blocks. These are the outcomes of decomposed software system focusing on modularity with key features. A set of closely coupled encapsulated components perform specific task with coherent functional logic.

Software testing in the Component Based Software Development is different from the classical development of software. The component source code is invisible and design make it very difficult to trace the faults while using the components of COTS[8].

The other new problems arises during the testing of such component integration that were made or used in another product. Sometimes the components may not work for certain requirements due to developed for another context. In such cases the fixing of problem after deployment increases cost of maintenance, waste of time and efforts[5]. The component providers of COTS will provide less information in the design document of the reusable component and do not reveal the complete information about the behavior of component in the other environment. These are the new problems arise in the Component based software development compare to the traditional software engineering[9]. The software testing teams implement strict testing criteria and various strategies to overcome these problems of invisible code of reused components. To ensure quality of overall product testing of each component is mandatory. But problems arise during testing, when the tester has limited knowledge about the component[3].

The software quality can be maintained throughout the process of software development and manual, semi-auto and automated tools which can simplify the testing process[3]. Software testing can be motivate the quality assurance of product in terms of economically and effectively viable for the large and small scale systems. The testing phase verifies the software quality before deliver to customer, whether it is really works as per the given requirements and specifications[10].

The overall quality of component based software product depends upon the quality of components which are assembled. Some of the following component behavior is relevant during the testing process.

Understandability : The documentation should consists content information with more clarity for use.

Controllability : The component characteristics of relevant with its input/outputs, functionality and services.

Observability : The component can be observed in the terms of operational behaviors, input parameters and generated outputs. The design model of component integration plays the vital role in determining the observability.

Traceability : The component facilitates the customization of its tracking functions.

Testability & Supportability: It is validated during the component test process and focuses during component validation.

The software testing experts believe that testing is a vital role in software development. Testing is parallel with other phases to deliver the quality product within right time. Testing teams interaction with developers to eliminate the bugs from initial stage of development. This process rectifies the bugs at the elementary stage rather than later stage, which reduces the overall cost of the product and reduces the testing time[12].

III. THE OVERVIEW OF COMPONENT TESTING SPECTRUM, AND ITS STRATEGIES

The Component Testing spectrum examines difficulties, challenges and various testing strategies of Component based software development[11]. The component based systems testing is more complex due to the invisible source code.

A. Complexities of Component Based Software Testing and Maintenance

- Component is independent module developed by third party for specific purpose and available at COTS. The component is very difficult to test due lack of source code only specifications are available. White box testing is problematic to test the structure of internal logic.
- Components are generally heterogeneous developed with different programming languages, executed in various operational platforms and architectures.
- Black Box testing is possible in the component testing. But sometimes even this test is also difficult to adopt due to lack of component knowledge that must be known for the software system.
- The software quality of component based software depends on the quality of components used to built it. The reliability test is mandatory in the component based software when new components are accommodated in the existing software for better features.
- The component reusability improve the software product quality and productivity. The component modification and upgradation motivate the maintenance
- Repetition of Testing during integration of components.

B. Component Testing Challenges

There are some challenges and difficulties in component based testing . The facts may cause the problems is as follows.

- The glue code of consumer components causes limitation for testing the components.
- There are some tools give support in component testing, but there no integrated testing environment which provides all features.
- The limitations to create test suites for generic test of similar behavior of components with different usability.
- Lack of Communication gap between user and the developer of component, the third party component may cause difficulties to test used for new product.
- Consumer and developer are not provided with component information for the unit test suites.

C. Component Based Testing strategies.

The testing teams uses various strategies makes easy to the testing and selecting the proper component for the developing the software product[6]. There are various strategies for the component based testing shown in table 1.

Table 1. Strategies of Component Based Testing.

Strategy	Testing Criteria
The Component Meta-data way	The component developer provides the additional information of the component such as control dependencies, complexity metrics, abstract view of source code and built in test suites. The meta data provides guidelines to component testing process.
Component Interaction Graph	Detect faults in the interfaces and interaction among the components. The Component Interaction Graph shows the data flow among the components and provide the general information of test cases of the components.
UML Based Test Model	The UML diagrams find out the faults of component interfaces using the Sequence and Collaboration Diagrams.
Built in tests in Components.	The method describes the test functions of the source code as a member function of the product to motivate the maintainability of the product.
Component Interaction Testing	This technique involves the each component assumption and how the component reacts with other.

These testing strategies implemented on the component based software at right time for test result.

IV. THE COMPONENT SOFTWARE TESTING TECHNIQUES

Effective software testing find out the high probability of faults and errors. The software engineer should develop the software product with “testability” in mind[7]. The effective test must exhibit the set of characterizes that achieve the goal of finding the more errors with minimum effort. The effective software test should have the following characteristics.

- The test should be specific and its own objective
- It should generate high probability of test errors.
- The test is not to be redundant
- The test should be moderate size for implement.

The software testing strategy should be flexible enough to promote a customized testing approach[8]. It must be rigid enough to promote reasonable planning and management tracking as the project progresses. It is useful process of executing program with the intent of finding bugs.

A. Adequate Testing

An adequate test suggested by David S. Rosenblum for the component based software product. The method is initial basis for component based software testing. The result of this method is formal definition of the concept. c-adequate – for p for adequate unit testing of the component and the concept c-adequate –for m for the adequate integration testing for the component based system [8].

B. Integration Testing

The integration testing is a rigorous in the form graphical representation of integrating black and white box information. The specification of each component is mentioned as component state machine which are similar representation of state machines in theory of computation[8]. Representation of Components mentioned with circles and dataflow from source to target state mentioned with arrows. Transition consists of five tuples such as source, target, event, guard and action. In this source and targets are the states, event causing transition, a predicate guard has to be satisfy before the transition performed. This method is a time consuming process with lack of automatic test case generation.

C. Self Testing of Component Software

The conventional testing methods are not adequate for the component based products. The self testing introduces the framework for the automation of customer oriented component testing that reduces the testing cost. This test is basically similar to black box testing and uses common features of commercial tools. The invention of this approach is the framework that enables automatic generation of tests without involving the knowledge of domain and source code. However it will highlights the weakness if any change happens in Component Based Software.

D. Automated Software Robustness Testing

The Robustness Testing technique is an automatic testing of software components robustness. I will verify the ability of component to overcome the invalid input conditions and how the software entity works under peak load environmental conditions. The test watches the problem of machine crashes, abnormal terminations etc. This approach targeted to focus the large no. of test outcomes generated from the small no. of test inputs. It reduces the number of test cases to be executed without test accuracy and reliability[8]. With this method, the developer can decide whether to release the software into market place or not, and the end user will decide whether to adopt that software or not. Based on reliability information the risk also measured. Robustness testing and stress testing are the variances of reliability testing.

E. Boundary Value Analysis

The representation of The Boundary value analysis mentioned equivalence portioning of input data into equal parts and then test cases are to be generated. This method designs the test cases and verifies the cases near to the limits of valid ranges[8]. Boundary value analysis generates test case that highlights faults better than equal partitioning. In automated regression testing, large number of test cases generated but the demand is to reduce the test cases.

F. Object Oriented Component Testing

The Object oriented testing has been derived by Fakhra Jabeen et al. In this approach testing frame work is proposed that relies on the usage of descriptors to prove the test execution and to enable uniformity of information flow. The component developer or third party tester has responsibility for descriptor unique to their perspective and to avoid ambiguity of missing testing information.

The approach supports unit testing of component and partial integration testing.

G. Event Flow Model

In modern days most of the software designed with Graphical User Interface which provides the usability, flexibility and user friendly nature to the user. The functional correctness of GUI ensure the robustness, safety and usability of entire features of the software product. Graphical User Interface organized with windows in hierarchical nature of the system[8]. GUI decomposed into a hierarchy of model dialogs. This hierarchal structure is an integration of modal dialogues follows the nature of flow graph that shows all possible event execution dialogue paths. Event flow model used to perform various testing tasks with model based techniques of event forest algorithm. Even forest structure generates various kinds of test cases automatically covering number of coverage's on criteria.

H. Regression Testing

Regression Testing confirm that the source code change has not adversely affected existing features. It is safe guarding procedure to validate and verify the adapted component, and assure that no errors have been incorporated. The source code full or partially selected which already executed, test cases which are re-executed to ensure earlier functionalities[8]. The regression testing is expansive when the tester needed to re-execute all test cases against the modified component Regression test process involves two main steps: (1) Identifying and highlighting the modified segments of the system, (2) Ensure the results which are not affected original features.

V. CONCLUSIONS

The component based software is a integration of individual components of specific task. The quality components motivate the quality product which depends on various testing strategies and methods of testing. The Testing complexity of Component based software more than the traditional software due the lack of source code. Software testing is crucial for software reliability and robustness. In this paper most prevalent strategies with different nature and objective are described. The advance research in component selection, design, implementation and validation are obviously important, but none of them will be ultimately useful without skilled practitioners to distinguish between good and bad. The some of the tools give support in component testing for specific purpose, but there no integrated testing environment which provides all features Still lot of research work to be needed in component software testing. It leads to deliver the quality component based software product

ACKNOWLEDGMENT

We sincerely acknowledge to all the faculty of the Department of Computer Science & Engineering, Indur Institute of Engg & Technology, Siddipet for their support and cooperation in the period of study.

REFERENCES

1. Asif Irshad Khan et al, "An Improved Model for Component Based Software Development", Scientific Academic Publishing, Vol.4(2), 2012
2. Bhupender Yadav et al, "A Review on Component Based Software Engineering and Testing", International Journal of Recent Advances in Engineering & Technology, Vol.1, 2016.
3. Cirstian Cadar et al, Symbolic Execution for Software Testing in Practice – Preliminary Assessment", ICSE, May, 2011.
4. G. Bernet, L Bouaziz and P. LeGall, " A Theory of Probabilistic Functional Testing", International Conference on Software Engg. Proceedings, 1997.
5. Gerald Kotonya et al, "Towards A Classification Model for Component Based Software Engineering Research", Proceedings of EMROMICRO Conference "New Waves in System Architecture, 2003.
6. Mohsin Imran et al, "Component Based Software Testing Strategies to Develop Good Software Product", 1st International Conference on Emerging & Engineering Technologies, March, 2014
7. Mohd. Ehmer Khan, "Different Forms of Software Testing Techniques for Finding Errors", International Journal of Computer Science Issues, May, 2010.
8. Neelam Siroh and Anshu Parahar, " Component Based System And Testing Techniques", International Journal of Advanced Research in Computer and Communication Engineering", Vol.2(6), June, 2013.
9. P. Brooks, B. Robinson, and A.M.Memon, "An initial characterization of industrial graphical user interface systems", Proceedings of the 6th IEEE International Conference on Software Testing, Verification and Validation, Washington, DC, USA, IEEE computer Society. 2009.
10. U. Farooq, C.P. Lam and H.Li, "Towards automated test sequence generation", in Australian Software Engineering Conference, 2008, pp.441-450.
11. Vikas Verma et al, " Introduction to Component Based Testing", International Journal of Advanced Research in Computer Science, Vol.2(2), May, 2011.
12. Y.Kim, M.Kim, and N.Dang. *Scalable distributed console testing : A Case study on a flash storage platform*, in *ICTAC-2010*, pp. 441-450

AUTHORS PROFILE



Jogannagari Malla Reddy obtained M.Tech(CSE) from JNTU, Hyderabad. and awarded Doctor of Philosophy in Computer Science & Engineering from Lingaya's University, Faridabad. At present working as Professor and Head of the Department in Computer Science & Engineering, Indur Institute of Engineering , Siddipet, Telangana State, His area of specialization in Software Engineering, Object Oriented Analysis Design, Data Base Management Systems and Management Information Systems. He published various research papers on information systems in reputed National and International Journals & Conferences



system areas.

Dr. S.V.A.V. Prasad done his PhD from Andhra University, presently he working as **Professor & Dearn(R&D)** in Lingaya's University, Faridabad, has 30 years of experience in Teaching and R&D. He published various research papers in National and International reputed Journals & Conferences. He guided many students in research programmes at university in communication system and information



Kothuri Parashu Ramulu completed M.Tech (Computer Science & Engineering) in 2010 from JNTU, Hyderabad and pursuing Ph.D(Computer Science) from Rayalaseema University, Kurnool, A.P. His area of specialization in Database systems, Software Engineering and Programming Languages. He published various research papers in National and International journals in the area of computer science.