IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-5, Issue-12 E-ISSN: 2347-2693

The Complexity of Verification and Validation Testing in Component Based Software Engineering

Jogannagari Malla Reddy[1*] and Kothuri Parashu Ramulu[2]

[1*]Dept. of Computer Science and Engineering, Indur Institute of Engineering and Technology, Siddipet, India
[2] Dept.of Computer Science, Rayalaseema University, Kurnool, India

[*]*Corresponding Author: jmrsdpt06@gmail.com, Tel.: +91-94924-79708*

Available online at: www.ijcseonline.org

Received: 09/Nov/2017, Revised: 22/Nov/2017, Accepted: 08/Dec/2017, Published: 31/Dec/2017

Abstract—There is a huge need of software professional to develop the better quality systems in rapid time to satisfy the customer needs. The component based technologies fulfil the achievement to make this possible. The modern systems have practically too complex behaviour. The complex systems developed with reusable components has many features such as lower costs and shortened development lifecycles. The component software development places the significant challenges like system integration and testing to build the software products. The verification and validation techniques are essential to ensure the software quality for component based producers. Accomplishing these techniques in the reusable component building is not a simple task. This paper focuses the various difficulties of these testing in various stages of component based software engineering.

Keywords—Requirement, Design, Construction, Deployment, Software Quality, Verification, Validation

I. INTRODUCTION

The software technology plays an important role from the past three decades making world as digital with its rapid emergent technology. The software is the youngest industry compared with other industries, but now it is nucleus for other industries. The industry demands quality products within short span of time and required functionality. The lack of quality leads the significant costs in time, budget and wastage human efforts for developer and unsatisfaction to the customer. The error tolerant systems can't fulfil the functional, non functional and domain requirements of the stake holder. The software industry investing 30-50% development cost on testing to making the systems as reliable.

The testing is most widely used technique to ensure the quality in the manufacture, electronics, communication and software industries where the innovative, redesign takes place to achieve the customer satisfaction and reach the market competitive nature. The software professional are paying more attention on testing in the areas such as control systems, communication systems, embedded systems and other relevant software paradigms

The software industry shifted from traditional software development to Component Based Software Development to develop the software systems in rapid manner with advanced methods. It is an emerging technology that, strives to construct high quality software with minimum budget in the short span of time. The Component Based Software Development can build the complex systems with standard pre-built components using with reusable concept. The component based approach has significant success in many applications domains such as web based and desktop graphical applications. The components developed in open market for reuse that will be used in many different configurations and situations, many of them not foreseen at the development time. The component developers delivered as black boxes [1] without source code.

In the traditional software engineering, the verification and validation could be done with customer co-operation and entire code is visible. The verification and validation of Component Based Software systems becomes a little bit more complicated than the systems built using traditional methods. The verification and validation (V&V) is forefront of software testing and plays vital role for development of quality products. This paper highlights the state of art of verifications and validations techniques in Component Based Software Development and representing the future directions in the research domain.

The remaining part of the paper is organized into various sections. Section 2 focuses on the related work of verification and validation. Section 3 describes the various phases of Component Based Software Development. Section 4 illustrates the role of verification and validation in the Component Based Software development. Finally a discussion about Conclusions and future scope is given in the section 5.

II. RELATED WORK

Over the years many researchers and theorists identified the vital role of verification and validation testing in Component Based Software Development and they targeted various key issues to improve the software product quality. These concept taxonomy can be useful to extend the knowledge base in the area of component based software engineering.

* I. Pavlova [1] identified the object oriented properties in component based software engineering such as encapsulation, information hiding and inheritance etc.

* Ishita Verma[2], emphasis the design and construction and applicability of component based software engineering and applicability of components with its advantages and disadvantages.

* Ivica Crnkovic, et al[3], described the various stages of Component Based Software Development and analyzed with case study of the company

* Peter Henderson and Robert Walters [4], elaborated the role of validation in Component Based Software Development. They briefly discussed the RolEnact tool implementation in the component based systems.

* Abhikriti Narwal[5] implemented the metrics on the component based software systems to improve the software product quality. They most of the finding got through the interviews of designers, developers and testers of the software industry. The research suggests various suggestions and recommendations for making software product as robust.

* Davis [6] described the various strategies of requirement Analysis relevant with stake holder needs and limitations of human information processers.

* M. Barjktarovie [7], illustrated the formal specifications and verifications through automated tools for design validation of component software development.

* Sullivan and J.C Knight descried the importance of software component reusability with architectural approach [8]. He discussed the complexity of verification and validation testing in the component based software engineering with glue code.

III.THE VARIOUS PHASES OF COMPONENT BASED SOFTWARE DEVELOPMENT

Component Based Software Engineering is advancement in the area of Software engineering. The various principles, procedures, methods and tools will be used similar way in CBSE. The Domain Engineering emphasizes with aim of finding functional, behavioural and data components for reuse which are stored in the Capability Knowledge Base (CKB) and reuse libraries. The area of domain engineering targeted to three major activities like Analysis, construction and dissemination of components for existing and future use. Component Software development focuses on the proper component identification, verification and forming the relations between them. Implementation effort is less required in the component based software development than selecting, locating the proper component and integration which saves the time and effort.

The development model starts with requirement specification progresses with system specification, selecting the proper components by evaluation and ended with integrating the components together in Component Based Software Development which consume the time and human efforts. There is a need of component adoption and testing before it can be integrating into the system [3] . The various following steps involved in the Component Based Software Development.

Requirement Analysis and specification: The requirement Analysis phase is primitive phase of CBSE same as Traditional software development life cycle. In this phase the requirement engineer can analyse, negotiate the requirements fulfilled with available components which are developed by the third parties. The appropriate the components always to be found, otherwise there is risk with new components have to be implemented.

Software design: The requirement specification phase and the design phase are strongly interrelated to the availability of components. Every component is complying with independent particular component model. It is practically difficult to use with another component technology to achieve the interoperability. The specific component requires suitable component model with comfortable architectural framework. This has been direct impact on architectural decisions. So, the design process highly coupled with its components [3]. Selection of suitable components which are fit for the architectural design may be sometimes difficult to find. There may be some gaps between the features of the component and software requirements. The good design usually needs more iteration. At the end of the design phase, the components are identified to build the application and new components that are needed to be modified for interoperability.

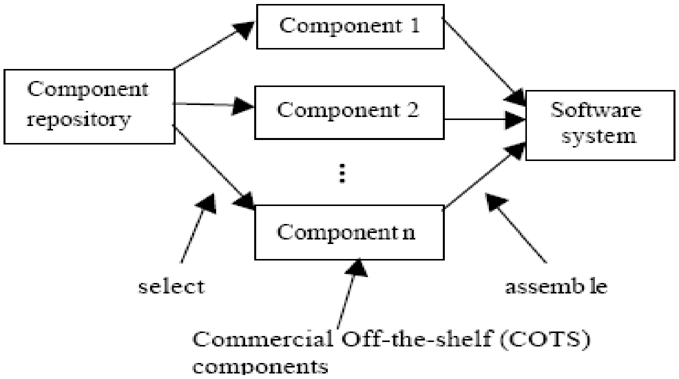

Figure.1. Component Based Software Development

Implementation and Unit Testing: The selected components are assembled, tested and debugging can be made as part of the implementation and testing. The "glue code" provides the interface among the components and implementation of new functions. The testing of components in isolated is not sufficient. The process of integration testing of components and possibility a glue code is more important than isolated testing.

System integration: Integration of components can form the subsystem of a application as a whole. Integration of particular components into a system called a component deployment.

System Evaluation: The developed software system is not be satisfied by the stakeholder in its life time. There will be process and technology changes occur in the business environment. Definitely it needs maintenance to full fill the client new needs. In most of the cases the existing components will be modified or new component will be integrated into the system.

The component based software engineering have the following advantages over the Traditional software engineering [2].

1. The component reusable feature reduces the development cost than the traditional software development

2. The development time will be reduced with component reuse than the traditional software engineering. There no need of writing the code in the component based software engineering which developed by third parties.

3. The component based software development applicability more with pre built components. There is no such restriction with traditional software engineering.

4. The components are developed and tested by the third parties which exhibit and assured the quality s like reliability, usability and performance etc. So, component based software development exhibits the assured quality than traditional software engineering.

IV. THE ROLE OF VERIFICATION AND VALIDATION IN COMPONENT BASED SOFTWARE DEVELOPMENT

This research concept highlights the verification and validation techniques with respect to the Component Based Software Development in an innovative direction. Before that, let us have to get the idea on the technical concepts of the related area of the topic.

The software testing is an activity to evaluate the effectiveness and capability of software product which meets the required results. Testing is series of activities that can be conducted in planned in advance with systematically. The software testing can made the software product with more reliability and robustness The implementation of proper testing strategy at the right time will returns the more efficient and effective results.

The component system developers require the verification and validation techniques to ensure the software quality. The component requires less verification than custom software systems. But if the quality of the used components is not enough, defect finding in component based systems is more difficult and expensive. The Component Based Software Development verification is not as not like traditional software development due to the un availability of source code the software faults can't be found and corrected with traditional debugging techniques. The advanced research techniques have to be used to find out the faults and workarounds has be implemented to avoid them.

In traditional software development the verification and validation made with close cooperation with stakeholders and their requirements. The Component Based Software Development verification and validation is little bit complexity compare to the traditional software systems due to invisible code. The impact of the malfunction lies on another component. Interface between the components plays main role in checking correct data of input and output of the component. The problem caused with incompatibility between components, or by broken dependencies. When the components are poor quality then verification of the software is very difficult task.

The Component Based Software Development applies the verification and validation techniques based on the following perspectives.

Component developer perspective: The component software developer can perform the rigid verification and validation of the component against the specification before delivering into the open market. The white box testing can rectify the errors in programming such as path, decision structures, control statements, and application of data structures compliance with the design. Finally, the functionality and quality of the component has be tested as block box against the specifications then it can be certified and delivered in the open market for reuse[9].

Component Consumer perspective: Components (Off the Shelf) are available as executable “black boxes” with licenses [10]. Due to this, the consumer can’t perform white box testing. If COTS component without certificate or certificate not trustable the black test to be performed then certify it for usage.

The component requires three types’ testes as part of verification and validation for its certification. Black box testing conducted based on system operational profile and System level fault injection test will be conducted to validate the system behaviour due to failure. Finally, operational testing is to validate the component perfect behaviour.

Significantly, the component based process kept less efforts on programming than verification testing. The verification activity performed at the various stages of the component software development with specific goals.

- Verification of the component in isolation
- Verification of the components in integration.
- System verification when a component has been deployed into the system.

Correctness of the complete system is verified and the system is validated with respect to the requirements. Validation is process of tracing requirements with final complete system i.e. after coupling of components together. The designer implements various methods and tools which are able support for this activity. The verification and validation in component software development differ with traditional software development in terms proofing, certifying, functionality and quality of software. The various techniques such as manual, tools based inspections, tests and live tests can make verification and validation more easy to develop the software product as robust [10].

V. Conclusion and Future Scope

The Component Based Software Development will reduce the cost, time and efforts than the traditional software development. There are some critical issues in proper selection and verification of components from the Capability Knowledge Base repositories. Improper integration leads to side effect on the other components of the system. The high quality software is achieved through verification and validation, which rectify the failure nature of software product. The various inspections, auto and semi automated tools can provide support for verification and validation testing in the Component Based Software Engineering. There is need of extendable knowledge in the area of verification and validation for developing the Components Based Software systems with more quality.

Acknowledgment

We sincerely acknowledge to the faculty members, Department of Computer Science & Engineering, Indur Institute of Engineering and Technology, Siddipet for their support and cooperation during the research.

References

[1] L.Pavlova, “*Testability of Component Based and Service Oriented Systems*”, Proceedings of 12 International Conference Automatics and Informatics, 2008.

[2] Ishita Verma, “*Study of Component Based Software Engineering*”, International Journal of Computer Science & Communication Volume V(3).2008.

[3] Ivica Crnkovic, et al, “ *Component based development process and Component Life Cycles*”, Journal of Computing and Information Technology, pp. 321-327, 2006.

[4] Henderson,P and Walters, “ *Component-based Systems as an aid to Design Validation”*, 14th IEEE International Conference on Automated Software Engineeiring , pp. 303-306, Oct, 1999.

[5] Abhikriti Narwal, “*Empirical Evaluation or Metrics for Component based Software Systems*”, International Journal of Latest Research in Science and Technology, Vol. 1, Issue. 4, pp. 373-378, Nov, 2012.

[6] Devis, G, “*Strategies for Information Requirements Determination*”, IBM system Journal, Vol. 21, No. 1, 1982.

[7] M. Barjktarovie, *“Formal specification and Verification of Communication Protocols Using Automated Tools* “, 1st IEEE International Conference on Engineering of Complex Systems, 1995.

[8] K. Sulllivan and J.C. Knight, “*Experience Assessing an Architectural Approach to Large Scale Reuse*”, 18th International Conference on Software Engineering, 1996.

[9] Vitharana, P, “*Risks and Challenges of Component-Based Software Development*”, Communication of the ACM, Vol. 46, Issue 8, pp. 67-72, 2003.

[10]. Christina Wallin, "*Verification and Validation of Software Components and Component Based Software Systems*", Extended Report of Building Reliable Component-Based Systems, July 2012.

Authors Profile

Mr. Jogannagari Malla Reddy obtained M.Tech(CSE) from JNTU Hyderabad. He is working as Head, Dept.of. Computer Science & Engineering, Siddiept. His area of specialization in Softare Engineering Object Oriented Analysis Design, Data Base Management Systems and Management Information Systems. He published various research papers on information systems in repured National and International Journal & Conferences. He has 15 Years of teaching exerpiecne and 4 years of Reserch Expericne

Mr. Kothuri Parashu Ramulu, Research Scholar, Department of Computer Science, Rayalaseema University, Kurnool, Andhra Pradesh. He is also working as an Asst. Professor in the department of Computer Science and Engineering, Indur Institute of Engineering and Technology, Siddipet. His area of specialization is Database systems, Software Engineering and Programming Languages. He published various research papers in National and International journals in the area of computer science.