

Testing strategies for stakeholders in Component Based Software Development

Khalid Mahmood* Bashir Ahmad

Institute of Computing and Information Technology, Gomal University D.I.Khan, Pakistan

* Email of the corresponding author: khalid_icit@hotmail.com

Abstract

Like other methodologies, Component Based Software Development (CBSD) has become emerging software development paradigm due to selecting reliable, reusable and robust software components and assembling all these into suitable software architecture. In CBSD, more emphasis is given to select, test and to adapt new component into existing software architecture. If these activities are not performed properly then it will impact the functionality and quality of software. During development life cycles of software all stakeholders especially tester roles are involved to overcome the errors and reduce the defects rates, so they need proper guidelines. In this paper, authors propose a strategy via workflow chart which helps the stakeholders at various phases of software development life cycle. Moreover, this strategy leads to synchronize the work of tester role and ensure the quality product on time

Keywords: CBD, Tester, functional testing, stakeholders, quality, CBSD, Third party.

1. Introduction

Quality and functionality of Component Based Software's can be enhanced if tester follows a proper guide lines. Like other methodologies, Component Based Software Development is an emerging method of software engineering (Nasib 2006). The basic idea behind the CBSD is to select, test and adopt the new components into existing software architecture according to well defined rules. In CBSD selection of new components is on the base of user requirements, integration testing of new components to check the proper working of software and finally implementation of whole component based software. In Software development life cycle software testing is considered as the longest and parallel activity, which is carried on, besides other activities. If this activity is not carried out in proper sequence and guidance it leads to software products with low quality and less functionality (Kai-yuan 2005). The author is emphasizing at component based system testing through work flow charts by ensuring explicit testing at each phase of software development life cycle. This will leads to synchronize the work of tester role and ensures component's reusability, dependability and safety.

1.1 Component-based Software Development Overview:

Component-based software development (CBSD) is an emerging field of Software Engineering (wiki 2007) that based on software reuse. It turns up from the limitations of object-oriented development to support effective reuse. Components offer a service without regard to where the component is executing or its programming language. A component is an independent executable unit that can be made up of one or more executable objects. The component interface is published and all interactions are through the published interface

1.2 Software Testing

Software Testing is a technical and independent investigation conducted to provide stakeholders with information about the quality of the component under test, with respect to the context in which it is intended to operate (Ram 1999). Software testing is the process of validating and verifying that a software system meets the business objectives and specifications, so that product works as expected and can be implemented with the same characteristics. Software testing is a parallel activity that start early with software development life cycle. However, how to test the software components is a problem since the source code of the software components under test may not be available to third party user (Fernando 2006). The objectives of software testing are to improve software quality, V&V, reliability and acceptance. The life cycle of software testing process is shown through Fig-1. Here in this diagram activities are represented by ellipses and milestones by rectangles.

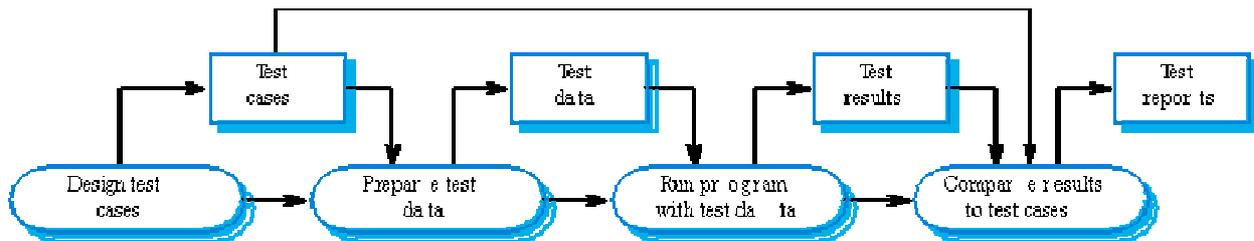


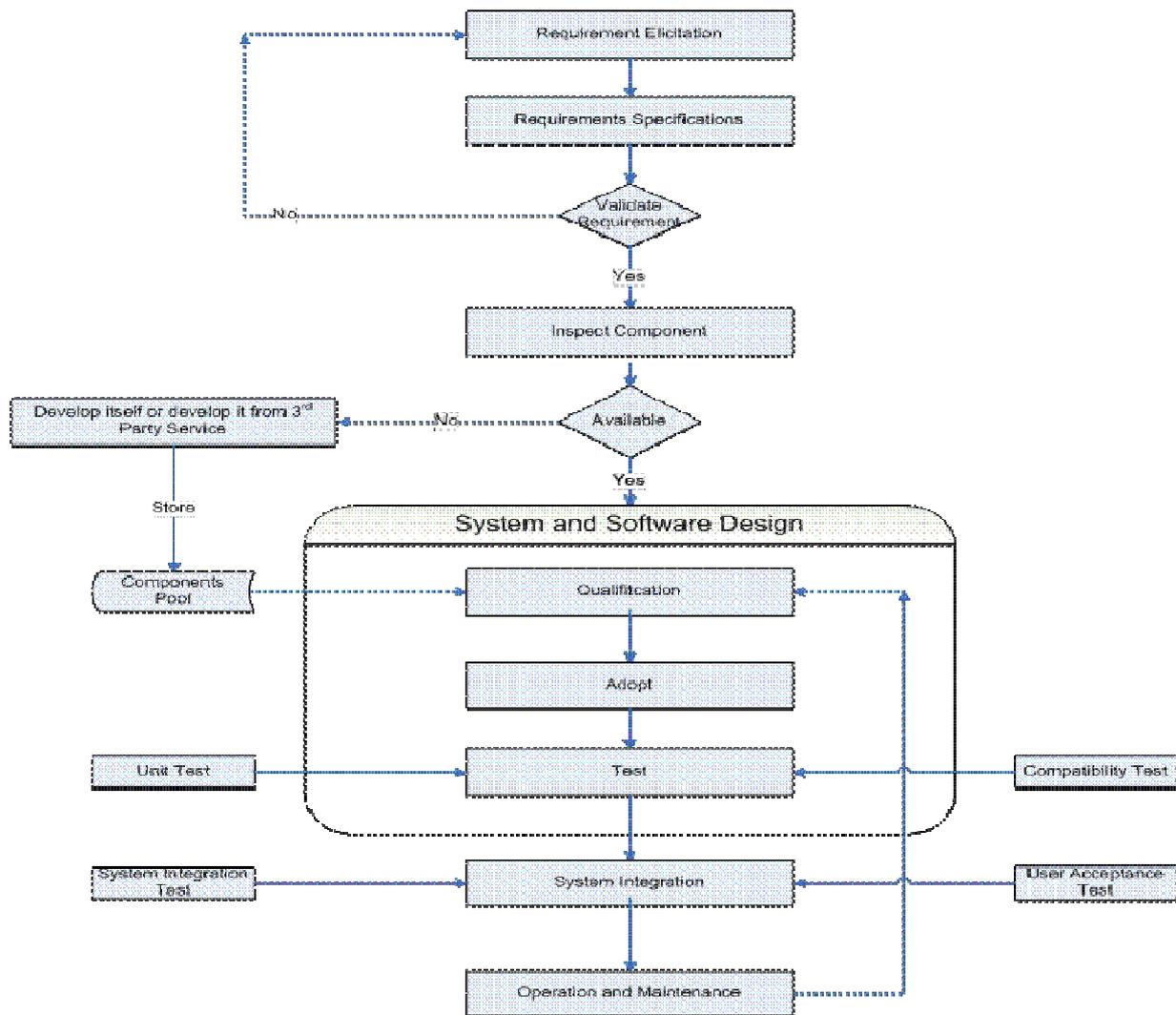
Fig-1. Activities and milestones of testing process

2. Component Based Software Development Life Cycle.

The activities that are carried in CBSD are Requirement Elicitation, Requirement Specifications, System and Software Design, System Integration, Deployment, operation & Maintenance. Requirement Elicitation represents the customer's need, constraints and goals that are to be achieved by the system. Requirement Specifications objective is to transform the user requirements into technical document that explicitly specifies the user requirements and system goals. System and Software Design activity elaborates the process of searching for components then evaluating those components according to requirements if they meet, if not then the requirements will be modified such that to adopt the available functionality of the component. If no such component exists according to requirements then these will be developed by third party services. System Integration describes performing System Integration test to check whether component is integrated and functioning according to the requirements stated. Deployment brings the component functional into existing system. Operations and Maintenance activity continues till the life of system in order to maintain the software according to new objectives.

3. Roles Involved in SDLC:

The author main concern is to provide guidelines to all stakeholders in Component based Software development life cycle to improve the functionality and capability of software as well as ensuring timely delivery of software. The stakeholders of SDLC are customers, system analyst and developers. Moreover, test teams including test manager, testers and other test team members. A proper guidance to these roles ensures quality product on time.



Proposed Workflow Model for Component-based Software Development

Fig-2: Workflow model for Component base Software Development

4. Proposed Workflow Model

The proposed workflow model presents the guidelines for all tester roles, to show the sequence of work and about the testing strategies which can be applied during different phases. The work of concerned tester roles and type of testing strategies are shown here.

4.1 Requirement Elicitation and Specifications:

In CBSD, the Software development life cycle starts by gathering and analyzing the Requirements by the System analyst who work closely with the customer to understand requirements, constraints and goals. Requirement specifications document is a technical report about specification of the user requirements either formally or informally. Validating the customer requirements early during the life cycle will reduce the error rates. In this step of proposed strategy, authors emphasis on validating the requirement specification document. This is carried on through technical reviews by concern team, inspection or walk through to confirm that requirement specification document actually demonstrates the user requirements, if it fails to depict the stated requirement then analyst can review the document through checklist to ensure that requirements are well understood.

4.2 System and Software Design:

After understanding the requirements the development team inspects the component according to customer's requirements. The availability of new component can be from three sources, from available components pool, from third party or developed from scratch by performing design, coding and testing activities. From available candidate components, a component is selected which will be compatible with user requirements, it is adopted and unit test is performed to ensuring the functionality, reliability, compatibility of the component. If the components are provided by the Third party then black-box testing is performed because the internal data structure and functions are hidden. But if the component is developed internally within the organization then all type of testing method are performed to test the components functionality and architecture of system.

4.3 System Integration:

After selecting and unit testing process, new component is integrated into the system by applying integration testing. This testing method will be applied to check whether the new/modified component works according to requirements or not.

4.4 Deployment:

Before the deployment process, system passed through alpha, beta or user acceptance testing process. After successful completion new system is deployed.

4.5 Operation and Maintenance:

The length of this phase mostly depends on the testing strategies which are adopted during other phases. If testing process in other phases done well then no extra time will be consumed during this phase.

The whole work of proposed work-flow model is summarized and shown in Table-1.

Component Based Software Development	Testing Types	Expected Tester Role	When	Targeted Results
Requirements Elicitation	User Acceptance	analyst, business manager	During elicitation process	Requirements will be gathered
Requirement Specifications	1. Verification 2. Customer Acceptance test 3. Inspection & reviews	Analyst, business manager	After or during Requirement Elicitation process	Requirements will be formally or informally specified
System and Software Design	1. Unit test 2. Functional test 3. Black box test	Designer & Developers	After successful specification of requirements	Design, coding and testing of components
System Integration	3. System Integration test 4. Compatibility test	Independent test team, SQA member, internal tester role	After unit testing of new components	Integrated Components
Deployment	Acceptance test	Deployment manager, developers, configuration manager	After System Integration process	Reliable, functional and qualitative Software System

Table 4.1 Summary regarding roles and action during CBSD.

5. Conclusion

Integration of new component in an existing software system need synchronize testing process at various stages of component based development process. Often tester role ignore the some testing technique which causes to impact the functionality of new system after completion of CBD life cycle. In this paper authors proposed a strategy to provide guidelines to all tester roles of concern phase of CBD. By adopting this strategy the work of all tester roles become streamlines and synchronizes. This would lead to development of quality products on time.

References:

- Nasib S. Gill and Pradeep Tomar (2006), “CBS Testing Requirements and Test Case Process Documentation Revisited”, ACM Sigsoft, Volume 32, No 2.
- Kai-Yuan Cai et. al. (2005) “Adaptive Testing of Software Components”, ACM Symposium on Applied Computing
- Component Based Software Development Topic, 2007 [online] Available: <http://www.wikipedia.org>,
- Ram Chillarege (1999), “Software testing best practices”, Technical report RC 21457 Log 96856, 4/26/99.
- Fernando R. C. Silva, Eduardo S. Almeida, Silvio R. L. Meira (2006), “An Approach for Component Testing and its Empirical validation.”

This academic article was published by The International Institute for Science, Technology and Education (IISTE). The IISTE is a pioneer in the Open Access Publishing service based in the U.S. and Europe. The aim of the institute is Accelerating Global Knowledge Sharing.

More information about the publisher can be found in the IISTE's homepage:

<http://www.iiste.org>

The IISTE is currently hosting more than 30 peer-reviewed academic journals and collaborating with academic institutions around the world. **Prospective authors of IISTE journals can find the submission instruction on the following page:**

<http://www.iiste.org/Journals/>

The IISTE editorial team promises to review and publish all the qualified submissions in a fast manner. All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Printed version of the journals is also available upon request of readers and authors.

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digital Library, NewJour, Google Scholar

