

# Integration and Change Management in CBSE to Support Migration

Farah Shafique, Sidra Shahid

**Abstract**— Software companies are trying to develop the products that are reliable, having short time for development, cost effective and to handle the change in requirements on daily basis. All these requirements lead them toward Component Based Software Engineering (CBSE). Achievement of CBSE is Component Based Development (CBD). The CBD gives more reusability, flexibility and maintainability than others. The main problem in CBD is to trace the requirement where change occur, manage this change in existing system and integrate it with system. In the existing research, the migration issue did not resolved, due to this problem many other problems could occur. In this research I tried to solve these issues by integration of traceability and change management. A model is proposed in this research which goes parallel with three techniques i.e. Requirement Traceability, Change Management and CBD.

**Index Terms**— CBD, Reusability, Change management, Requirement Management, Requirement Traceability, Migration Issues, Integration.

## 1 INTRODUCTION

SOFTWARE is key part to perform any complicated task in any organization. In this modern era no organization can do their business without software

In modern age wide changes have occurred in Software Engineering. Software Engineering is a branch that keeps the software in an efficient, well-organized and quantifiable way. It is an approach that improves, functions and maintain the Software. It is the process of utilizing our knowledge to produce effective systems [7].

One thing that is constant in software is "CHANGE". To handle the changes different approaches are used. Component Based Software Engineering (CBSE) is also one of them. Component Based Software Engineering (CBSE) is an approach that is based on reuse. Key idea of CBSE is to build the systems form existing components. It is an active era of research now a day. In Component Based Development when we reuse the existing components some kinds of problems occur while migrating and shifting the code components. Such kind of problems causes many issues and disturbance during development. To solve such kind of issues is to manage the change. Tracing the origin of requirement and change management is solution of mostly problems in CBD [14].

Traceability and change management are the techniques that are used for requirements gathering. Traceability upturns the understandability in complex components and increase the understandability of dependencies among the objects of software development [13].

## 2 REVIEW OF LITERATURE

Software components are used from software repository which is used to store, manage and rescue the components. A software component is a self-contained and an independent part of a software system that have complete functionalities. These components are developed earlier and reused whenever required in new system and modified according to the new system. In this paper it is discussed that choosing and reusing the appropriate component is challenging task specially when we have a large group of components and little documents about how component could perform and how should be reused [9].

Component Based Development (CBD) plays an important role for the productivity of any organization. For this purpose organization require many set of components in repository so that these components can be reused when required. However availability of specific component, its integration and change that component according to the new requirement is challenging issue. The component based model starts with the proof of identity of the applicant components. This is attained by classifying the data that is to be changed by the request and different identified relevant algorithms are applied. Once user components are recognized, the depository is checked whether the required components are existent in the depository. If component is available, it is retrieved and reused [3].

Development of software and its delivery is evolutionary in Component Based Software Engineering. Reuse oriented software process make it possible to reuse the software. Two processes involved in CBSE; Component Development and integration of different components. Different organizations can do these two processes and it is also possible to perform these two processes concurrent. CBSE requires to rethink different aspects of development. Most of predictable methodologies like object oriented methodology adopt development from start and are not providing help for reuse-oriented [2].

In software engineering component based development approach is in leading edge phase now a day. Due to reusability

- Farah Shafique, Master of Science (Computer Science) University of Agriculture Faisalabad, Pakistan, PH-0332-7054165. E-mail: farahshafiq1211@gmail.com.
- Sidra Shahid, Lecturer (Computer Science), University of Agriculture Faisalabad, Pakistan, PH-0334-6665173. E-mail: sidrashahid\_uaf@yahoo.com

of components it is called reuse centered software engineering. It is more flexible approach than other two approaches i.e. structured and object oriented approaches. Component based development approach supports integration and help to select the appropriate component but components should be migrated and integrated very carefully to adjust any change [4].

To cope the requisite of industry from past few years different attempts are adopted. Component Based Software Development is also one of them. In this paper different aspects of Component Based Development is discussed. Each reusable binary portion of code is said to be a component. Main advantages are reliability, efficiency, reduced complexity and improved quality of product and main disadvantages are integration, adoptability, security and customization. Component based software artifact line reliability estimate model that is capable of providing the lower and upper reliability boundaries that guarantees for a software product line article model, its specialism's and alignments[5].

In Software Project Management planning and requirements management phase recognizes and have tracks of all agreed changes of software requirements, relationships and dependencies between these requirements. Requirements management is actually a never-ending activity. It can be performed even after development and during maintenance phase because requirements change continuously. System must have any criteria which gives response to the changing requirement and properly allocate it. The purpose of assigning allocation to requirements is view and manage information and handle dynamically changing of requirements at any level of abstraction [13].

Now a day's many software companies are moving towards developing the time saving, short term development, cost-effective and reliable products and it moves them in the direction of CBSD. In Component Based Development how to select the component is the most important part. The key problem is selection of the component on which the complete software depend. The incorrect selection of component may increase time of development, extra cost and may cause the failure of the main software [10].

Major challenges faced by software community is maintenance of complex and large-scale systems. New trend CBSE is adopted to overcome such kind of challenges. In CBSD different components such as third party component or in house components are integrated. But when we shift from traditional engineering to CBSD several non-technical and technical issues occur. Every software component when it is adopted must have the capability to be modifiable and adopt different functional changes [14].

Tractability is discussed extensively in the research area. In software engineering research is it a topic of interest? Tractability of requirement is one of the important issue in the documentation that need to be more research. The main purpose is to discover system requirements and change these requirements agreeing to the customer's need, at this point, these problems had a lot of issues due to the component base development, any misperception or ambiguity, and then to solve this documented requirements are collected and the related parties have to confirm with these requirements [1].

Evolution of software is not just the process of continually changing and upgrading the software but it is also an important characteristic of software. Component Based Software Development (CBSD) is a core technology for software reuse. For better management in CBSE this paper provides a framework and in this framework a number of components including change control, tracing and tracking are integrated by the significant change pattern which is based on a process engine. Besides the technologies, in component base software development, people's skills and appropriate tool is also needed [15].

The requirements of any software project is directly related with stakeholders. The well-organized projects play an important role like a catalyst in component based development. There is a lot of rules of component based development (CBD) which effects the development procedures and keeps it maintain but still these requisite for considerable modification to achieve standard values. In CBD the selection of reusable component and its integration is hot issue for development community. For the effective use of possible component, when stakeholder request for change, it is necessary trace that requirement properly to select the exact component from the warehouse [8].

### 3 MATERIAL AND METHODS

In component based development to handle the change management and integration I'll propose a traceability and integration model. In this model I'll try to solve the problem by two requirements gathering techniques i.e. Change Management and Requirement Traceability. Traceability is a technique in which requirements are map out properly as user stakeholder define or as he wants to define [11].

Requirements management is a constant process of the project development and it relates to recording, evaluating, tracing and arranging the requirements and then lastly controlling changes. Change management is a process in which requirements can be changed at any time and at any phase of development [10].

In proposed model I'll go parallel with these two technique to solve the issues. For the validation of proposed model a survey is conducted. A Questionnaire is prepared for the survey [12].

### 4 DEVELOPMENT OF PROPOSED MODEL

The proposed solution deals with Component based Software Engineering along with the Requirements traceability and Change management. In component based development to handle the change management and integration. The life cycle of proposed model consists of the phases of change management and requirements traceability. It is defined by certain practices to be made by working. It is presented by management of requirements in the field of requirement engineering. There are three phases of proposed model of Integration.

#### 4.1 Change Management

Requirements management is a constant process of the project development and it relates to recording, evaluating, tracing

and arranging the requirements and then lastly controlling changes. Change management is a process in which requirements can be changed at any time and at any phase of development [10].

## 4.2 Requirement Traceability

Traceability is a technique in which requirements are map out properly as stakeholder define or as he wants to define [11].

## 4.3 Component Based Development

Component Based Software Engineering (CBSE) is an approach that is based on reuse. Key idea of CBSE is to build the new systems form existing components.

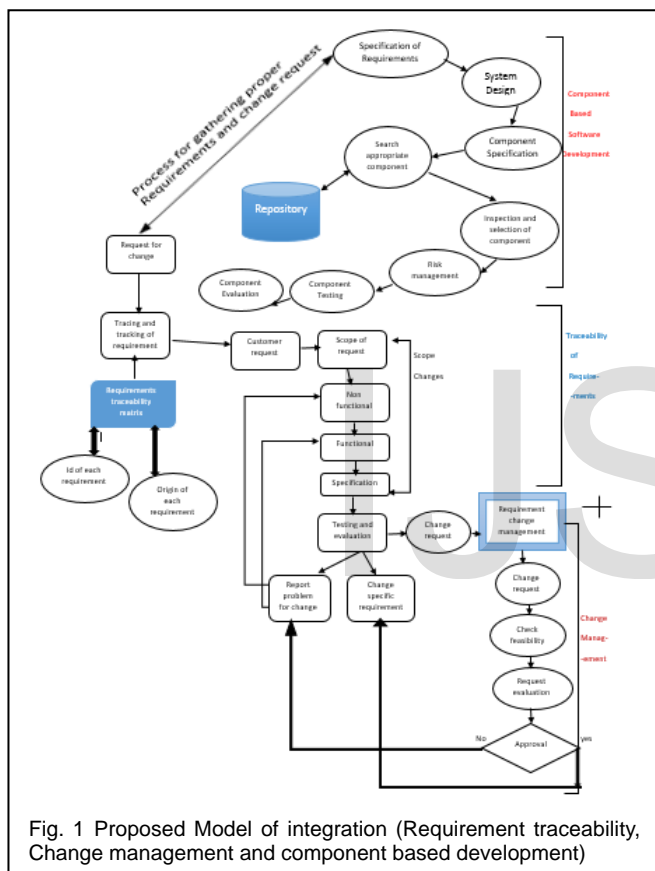


Fig. 1 Proposed Model of integration (Requirement traceability, Change management and component based development)

## 5 RESULTS

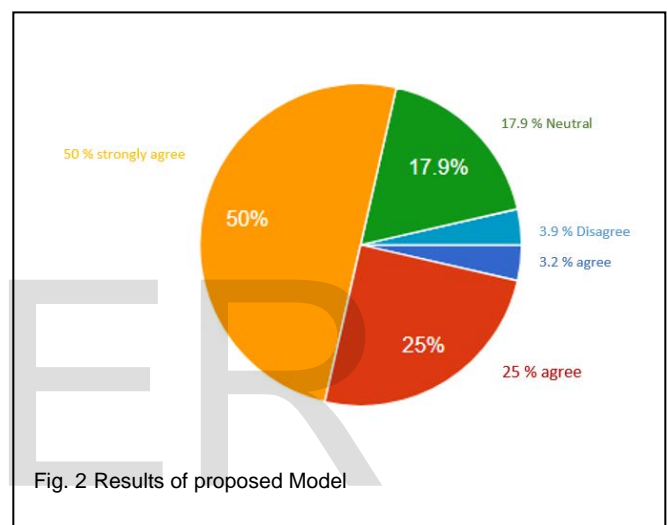
Verification of integrated and changed management model (new proposed model) is very important part of the research and its accuracy for the target group of users is also important point. The propose model is verified according to the following methods:

As a researcher I developed an online version of questionnaire for the validation of the proposed model and get response's of developers from different software houses. In such a way different experienced software developers from different communities participates in research and give their views about the model. They verified the given model by applying it on different small projects and give the views that they have faced some challenges during the development are solved as

much possible using this model. Their views and results are disused below.

I made a questionnaire about the component based development, change management and requirement traceability and distribute that in different software houses and ask them to give their opinion according to the market requirements. The respondents of the questionnaire were the IT support officer, web developer, project manager, block chain developer, software engineer and requirement engineer etc. In this way I receive reasonable answers and views about the model and it covers all the aspects about the requirement engineering. All their views and discussions are described below.

The results of questionnaire of how strongly developers recommend Proposed Integrated and Change Management Model are given below in Pie Chart.



## 6 DISCUSSION

Whenever the failure of any software is discussed the most common reason of failure is mostly the inappropriate requirements gathering, not understanding the view of end user, less interaction with the stake holders are considered. Till now a lot of work has been done in this field like elicitation of requirements, prioritization and documentation of requirements but till now it is considered more problematic area.

Similarly in component based software development identification of proper component, selection of component and its integration is considered problematic area. In the proposed model Integration and Change Management this problem is solved and try to understand where problem occur by keeping the record of all requirements in traceability matrix. In this way migration issues of code components are also resolved. In this model to migrate the code component it is first traced to its origin to check whether it is related or integrated to any other component or not. If it is integrated how to use it functionality on its own. So all migration issues of component based software development are solved in this model

## 7 SUMMARY

Software Engineering is a branch that provide an efficient, well-organized and quantifiable way for development. In this modern era every organization is doing best in development. . Where ever we talk about the reusability CBSD is considered the most important approach. CBSD improves the quality and productivity of development. It provides efficient, effective, cost effective and time effective way of development by reusability.

In Component Based Development some problems occur while migrating the code components in reusability. In this research these problems are discussed and their solution is provided.

A new model is proposed named INTEGRATION AND CHANGE MANAGEMENT Model. There are three phase of this model.

- Component Based Development
- Requirement Traceability
- Change Management

After developing the model it is validated by survey conducted from different developers of software houses.

## 8 FUTURE WORK

There are some points and aspects of the model which still needed to be modified and require future work. Due to unsatisfactory collected works and not proper tool support, some important activities are still overlooked. There should be a specific rule or requirements to choose the model. No environment is available for complete component based development

## REFERENCES

- [1]. Ayub, I., T. Mustafa, and A. Maqsood. 2015. Integration of Traceability and Change Management to Support Migration Issue in CBD. *Science International journal*. 27(5): 4379-4383.
- [2]. Azram, N.A., and R. Atan. 2012. Traceability Methods for Software Engineering Documentation. *International Journal of Computer Science Issues*. 9(2).
- [3]. Basha, N.M.J. and A. Moiz. 2012. Component Based Software Development: A State of Art. *IEEE Advances in Engineering Science and Management*. 599-604.
- [4]. Bakshi, A, and R. Singh. 2013. Component Based Development in Software Engineering. *International Journal of Recent Technology and Engineering. (IJRTE)* 2(1): 48-52.
- [5]. Bagheri, E, and F. Ensan. 2014. Reliability Estimation for Component Based Software Product Lines. *Canadian Journal of Electrical and Computer Engineering*. 37(2).
- [6]. Carmona, J, and J. Kleijn. 2013. Compatibility in a Multi-Component Environment. *Theoretical computer science*. 1-15.
- [7]. Gupta, M and D. K. Somwanshi. 2014. Selection and Implementation of Component in Component Based Development using Analytical Network Process. *International Conference on Green Computing Communication and Electrical Engineering*. 1-7.
- [8]. Islam, F. 2017. An Effective Approach for Evaluation and Selection of Component. *Computer Engineering and Intelligent System*. 8910:1-6.
- [9]. Kaur, V, and S. Goel. 2011. Facets of Software Component Repository. *International Journal on Computer Science and Engineering*. 3(6): 2473-2476.
- [10]. Khan, M.N.A., M. Khalid, and S. Haq. 2013. Review of Requirement Management Issues in Software Development. *International journal of modern education and computer science*. 5(1): 21-27.
- [11]. Lai, R, and N. Ali. 2013. A Requirement Management Method for Global Software Development. *Information and software technology*. 1(3): 38-58.
- [12]. Mathiyazhagan, T, and D. Nandan. 2010. Survey research method. *Media Mimansa*. 4(1). 34-82.
- [13]. Swarnalatha, K., G.N. Srinivasan, M. Dravid, R. Kasera, and K. Sharma. 2014. A Survey on Software Requirement Engineering for Real Time Projects Based on Customer Requirement. *International journal of advanced research in computer and communication engineering*. 3(1).
- [14]. Tiwari, A., and P.S. Chakraborty. 2015. Software Component Quality Characteristics Model for Component Based Software Engineering. *IEEE International conference on computational intelligence and communication technology, (CICT)*. 47-51.
- [15]. Zhong, I., J. Xia, and X. Huang. 2016. The Framework and its Implementation for Managing Component Based Software Evolution. *3<sup>rd</sup> international conference on information science and control engineering. (ICISCE)*. 711-715.