ISSN (e): 2250 – 3005 || Volume, 06 || Issue, 12|| December – 2016 ||
International Journal of Computational Engineering Research (IJCER)

'O' Model for Component-Based Software Development Process

Saurabh Rawat[1], Rajesh Kumar[2]
[1] Graphic Era University Dehradun- India,
[2] Graphic Era University Dehradun- India,

ABSTRACT

The technology advancement has forced the user to become more dependent on information technology, and so on software. Software provides the platform for implementation of information technology. Component Based Software Engineering (CBSE) is adopted by software community to counter challenges thrown by fast growing demand of heavy and complex software systems. One of the essential reasons behind adopting CBSE for software development is the fast development of complicated software systems within well-defined boundaries of time and budget. CBSE provides the mechanical facilities by assembling already existing reusable components out of autonomously developed pieces of the software. The paper proposes a novel CBSE model named as O model, keeping an eye on the available CBSE lifecycle.

Keywords: *Component Based Development (CBD), Software Development Life Cycle, OModel, COTS, Dependency among components*

I. Introduction

Component-Based Development (CBD) is straightaway well ingrained in the IT industry. A component is a well-defined unit of functionality with a distinct line that grants it to liaison up with other components, and be separately spread out. The Component-based functions are explicit from the assembling components.

The Component-Based Software Engineering (CBSE) has become outward [1] in the genesis of 1990's. Originally, the ultimate products developed by CBSE were circumscribed to PCs whereas the use of COTS (Commercial Off-The-Shelf Components) software has inspires CBSE for the evolvement of business applications [2].

The extensive interests correlated with component-based technologies blend: evolution of condensed system, brisk induction, lessened cost, amplify quality, and curtailed system evolution and diminished maintenance cost. The increment in time has given upswing to the development of standard component-based specifications and the significance of CBD has developed swiftly in the embedded system trade.

A component is related to be autonomous bit of software if it has an open interface, distributes clear performance and moreover gives plug-and-pay services. Therefore it can be declared that component-based software evolution advocates the reusability and gives improved software quality. Thus Component-based software development can point to new ideas for the construction of large and complex software systems.

II. Review of Literature

The diverse kinds of CBSD models are available in the industry as well as in the academic world. We referred to some of them, in this division some of them are examined that are as follows:

The concept CBSE has actually make the scene after the fruitful set afloat of Microsoft's COM+ [3], SUN 's Enterprise JavaBeans [4], and IBM Component Broker [5] and CORBA [6]. CBSE have burst in among the conventional software technologies [7]. Furthermore, cumulative transfer of software attributes or stages that consist of a software product line, is expected to be in the limelight in the imminent years, therefore component-based software engineering has conclusions for how software engineers achieve, gather and keep up software systems [8]. Thus, we should see extreme modulations in designers' primary roles and required skills for software development in the imminent time.

All different stages of the software process are revealing and figuratively depicted by a Software Life Cycle Model . The phases of the software development cycle [9] are represented by Software development life cycle (SDLC) model.

The Twin Peaks model [10] also offers for a parallel, around the clock evolution of requirements and architecture all through expansion. Software can be developed by partial and easy wave. .

In X Model, the mechanisms are initiated by essential engineering and requirement assessment. Reusability in which software is refined by dwelling reusable components and software evolution from reusable and testable components, is the main characteristic of this software life cycle model .Develop software component for reuse and software development with or without modification in reusable component. [11] are two main ways, it uses in software development.

Software reusability during CBSD is exemplified by the Y Software Life Cycle Model. The Y Shape of the model weighs up iteration and overlapping. The designed phases of Y model are: domain engineering, frame working, assembly, archiving, system analysis, design, implementation, testing, deployment and maintenance. [12], although the principal phases may overlap each other and iteration is granted.

Knot Model recommending on reusability, recognizing risk analysis and feedback in each and whole segment. This model Knot model is based on three states of the component [13], may be best matched for intermediate or larger complex system's development.

Expansion of new product using component based technology is accomplished by promising software lifecycle model, the Elite Life Cycle Model (ELCM). This model portrays a general process of Software development with the help of in built components. [14].

New age with new permutation in Software Development [15] gives the thought of selection, prioritization and customization to develop, transform and choice of components.

The V model adopted the conventional software development access for constructing a system from reusable software components [16]. It consists of several steps and supports the details information at the design phases. The central significance of V-Development is component development lifecycle. Component development lifecycle was regarded as contrasting process. The selection phase gets input from the independent system that usually finds and develops the appropriate components to be composed into the system. The rigid conventional waterfall model for modular system development with little flexibility is the V Model.

The W lifecycle model, amalgam of two V models together. Component based development process comprises of a component life cycle and a system life cycle, and it is the base of W lifecycle model [17]. The W model accomplishes all the concerns of component based evolution. The W Model entertains a V model for both component and system life cycles.

III. 'O' Model for Component-Based Software Development

This study propose a new 'O' model in which the processes start in usual way by requirement engineering and requirement specification as shown in Figure 5.1. In a non- component-based approach the process would continue with the unit design, implementation and test. Instead of performing these activities that often are time and efforts consuming, simply select appropriate components and integrate them in the system. However, two problems appear here which break this simplicity– Itis not obvious that there is any component to select, and the selected component only partially fits to our overall design. The main characteristic of this software life cycle model is reusability in which software is developed by building reusable components for software development, and software development from reusable and testable components. In software development, this research use two main approaches, develop software component for reuse and software development with or without modification in reusable component. Evolution and the production of potentially reusable components are meant to be useful in future software projects. Reusability not only involves reusing existing components in a new software system but also producing components meant for reuse. When a software system has been developed, the software engineer may realize that some components can be generalized for potential reuse in the future. Reusability implies the use of composition techniques during software development; this is achieved by initially selecting reusable components and assembling them or by adapting the software to a point where it is possible to pick out components from a reusable component repository and testable component repository. This study introduces two main phases first is building reusable components for software development and second is

building software from reusable and testable components *(Tomar and Gill, 2006)* of O model which help in developing a component-based software.

a. Phases of O Model

1) Component selection/ Modification/ Development
2) Integration of Components
3) Testing of Component
4) Customer/User Evaluation

b. Component Selection/Modification/Development

Once a build scope is established, we need to decide which of the required components can be used (e.g., already exist in the organization or can be bought off-the-shelf) and which ones need to be developed. And some components need modification (see figure 2).

3.1.1. Using existing component

Reusing an existing component may require some adaptation. For example, the component interface might not be exactly what is required or some of the method behaviors may need alteration. This is achieved through adaptation, which involves wrapping the component with a thin layer of code that implements the required changes.

3.1.2. Modify Existing Components (if required)

Developing a new module from scratch is always avoided in CBD. It may highly possible that some existing components may require some minor or major modifications to accommodate with other components. We can modify an existing component according to the scope specification.

3.1.3. Development of New Components

Building a new component should always begin with defining the component interface. This represents a permanent contract between the component and other components. Once the interface is defined and the intent of each method is established, the component can be designed and implemented.

3.2 Integration of Components

With all the components for a build in place, the components are then integrated and tested. The Integration process must be done keeping scope specification and design specification in mind. The integration of Elite model is based on *Clustering Approach*. We must start integration with bottom level and progresses toward the Clusters. These different clusters will ultimately form the software. Integration will require the writing of intermediate code that establishes the interaction between the components.

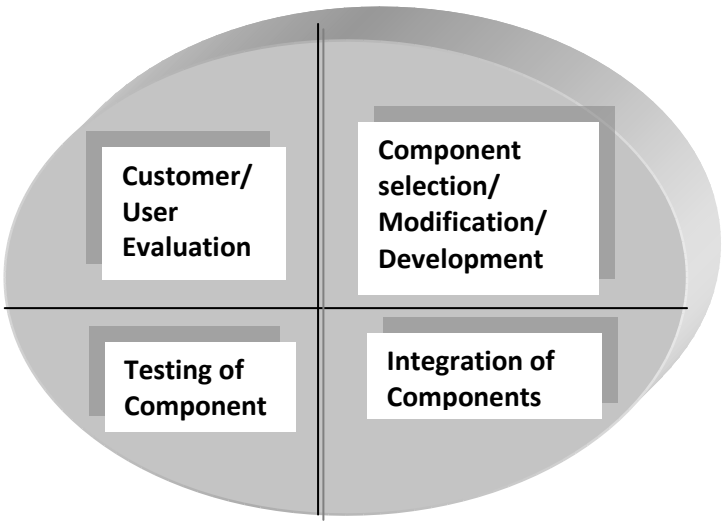

Figure 1: *O Model for Component-Based Development*

3.3 Testing

Testing is the most important activity of the software development process for finding the maximum errors; therefore without proper testing of the software product, all the efforts will be in vain [SZYP 99, MYER 04, and GILL 07]. Software Component Testing (SCT) is an approach to finding errors, reducing cost, improving reliability, and enhancing the quality of software components [MEYE 03, BEIZ 90 and MEYE 98]. In CBD, testing is applied not only to the individual components but also to the whole integrated software system. This process not only fulfills the aim of finding errors but also improves the software quality [SITA 94]. SCT represents a group of activities, which involves component study, quality test design and generation, test execution, fault detection, and finally testing evaluation [BEIZ 90].

1.4 Customer/User Evaluation

The evaluation environment may be the same as the development environment (for earlier builds that are not mature), or a pseudo live environment (for later builds that are sufficiently mature). The outcome of the evaluation influences the direction of subsequent builds. The Evaluation phase involves the answers to these questions:

- Is the user satisfied?
- Are the actual resources expenditures versus planned expenditures still acceptable or not.

VI. Conclusion

This chapter recommends a novel O component-based model for CBSD. O model for CBSD work out to rationalize the evolution of a software system with four major phases-Component Selection, Integration, Testing and Customer evaluation. There are several sub-phases - software analysis and specifications, design, coding and archiving, component testing, component wrapping, domain analysis, domain engineering, system testing, implementation and deployment, and maintenance of software components. O model helps in developing CBS with the help of two CBSD approaches, namely, development for reuse and development with reuse. Finally, O model appears to cover the likely phases of large software development and enforces software reusability along its phases. Likewise, it takes into account previous knowledge that software engineers may have about the application domain, which has an impact on the prevalent approach to be pursued during the software development with this model.

References

[1] C. Szyperski, Component Software, Addison-Wesley, 1998.
[2] M. Aoyama, Componentware: Building Applications with Software Components, J. of IPSJ, Vol. 37, No. 1, Jan. 1996, pp. 71-79 (In Japanese).
[3] Microsoft, 2004. COM+, http://www.microsoft.com/com/tech/complus.asp.
[4] SUN, 2004.Enterprise Java Beans, http://www.java.sun.com/products/ejb/index.html.
[5] IBM,2004. Component Broker, http://www.software.ibm.com/ad/cb.
[6] Object Management Group, 2004. The Common Object Request Broker Architecture, http://www.omg.org.
[7] Wallnau, K. C., S.A. Hissam and R.C. Seacord,2002. Building Systems from Commercial Components. Addison-Wesley.
[8] Clements, P. and L. Northrop, 2002. Software Product Lines. Addison-Wesley.
[9] S. Cohen, D. Dori, U. de Haan, "A Software System Development Life Cycle Model for Improved Stakeholders Communication and Collaboration", International Journal of Computers, Communications &Control,Vol. V (2010), No. 1, pp. 20-41
[10] Royce, W.W., 1987. "Managing the development of large software systems". Proceedings of 9th IEEE International Conference on Software Engineering, pp: 328-338.
[11] Gill N. S. and Tomar P., "X Model: A New Component- Based Model", MR International Journal of Engineering and Technology, 2008, Vol. 1, No. 1 & 2, pp. 1-9.
[12] Luiz Fernando Capretz, " Y: A new Component-Based Software Life Cycle Model ", Journals of Computer Science1 (1) : pp.76-82.
[13] Rajender Singh Chhillar, ParveenKajla, "A New Knot Model for Component Based Software Development", International Journal of Computer Science Issues Year: 2011 Vol: 8 Issue: 3 Pp.: 480-484
[14] LataNautiyal, Umesh Kumar Tiwari, Sushil Chandra Dimri, ShivaniBahuguna, "Elite: A New Component-Based Software Development Model", International Journal of Computer Technology & Applications (IJCTA), Vol 3, Issue 1, Jan 2012, pp 119-124
[15] LataNautiyal, UmeshTiwari, SushilDimri&Shashidhar G. Koolagudi, "Component based Software Development- New Era with new Innovation in Software Development," International Journal of Computer Applications (IJCA), vol. 51, no. 19, pp. 5-9, August 2012
[16] IvicaCrnkovic; Stig Larsson; Michel Chaudron, "Component-based Development Process and Component Lifecycle." Online Available: http://www.mrtc.mdh.se/publications/0953.pdf
[17] The W Model for Component-based Software Development [online]. Online Available: http://www.cs.man.ac.uk/~kung-kiu/pub/seaa11b.pdf.