ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 12(2), DOI: 10.17485/ijst/2019/v12i2/138330, January 2019*

Component Based Systems: A Novel Methodology in Software Engineering

Usama Quadri Syed*

Department of Computer Science, DCC, Shaqra University, Kingdom of Saudi Arabia; usyed@su.edu.sa

Abstract

Background/Objectives: The component based systems make the possibility to develop a better quality system in a faster way in order to satisfy the customer needs. Component based software development is the new technology in software engineering. Component based methods phases the importance of reusing the existing components in software engineering thereby, ensuring lower cost which leads to a versatile development approach. **Method/Statistical Analysis:** In this study, a novel methodology of Component Based Software Development, its various issues, advantages and limitations in developing software is proposed. This study also focuses on component-based software engineering and to reveal how it can be utilised efficiently and effectively to develop high-quality software products. **Findings:** Under ideal situations, with minimal testing efforts, integration of reliable software components should produce high-quality software. It additionally, it provides a short summary of the various factors; its models and additionally highlights a number of points regarding the performance and quality factors that are necessary for a software development. **Application/Improvements:** Feature enhancements in this can be a new technology that could overcome the existing drawbacks

Keywords: Component Deployment, Reliable Software, Object Oriented Software, Software Engineering

1. Introduction

A Component Based System (CBS) associates with the assembly of the elements of computer code, which follows software engineering method that is a Component-Based Software Engineering (CBSE) procedure[1]. Totally, different languages of programming will be used in writing these elements and those can also be used in various operational platforms, such as the internet. Number of these kinds of software can be proposed deployed by any third party developers, whose ASCII text file is sometimes not offered to the users except the usable file. By permitting the combination of reusable elements from totally different venders, the CBS overcomes these disadvantages and it also improves the development of quite advanced systems in an advanced approach. This approach doesn't have an effect on the specified ensuing system. It also provides a more effective and flexible environment for the dynamic computer code. At the identical time, various new issues have been introduced in the process of testing and maintaining this CBS based computer codefurther[2–5].

Initially, the CBSE emphasised on the computing based multi user. But, the employment of Ready-to-use package improved the overall process of CBSE within a different level of upgradation[1]. Also, fast development and its deployment in the web based technology like the internet and Java-based applications require new prospects of CBSE.

2. Component Based Software Engineering

2.1 Component of Software

A component is defined as the part of an object which is already made ready. In this study, it can be defined as a system. In the software-based engineering paradigm, this might enable software to possess as elements. The

*Author for correspondence

programming language directions all the sub-routines, algorithms, sub modules, computer code packages etc. The widely recognised aim of the component-based development is to create and maintain computer code systems by victimisation existing computer code elements. In[1,4,6,7] proposed that these systems have to act with one another in designing a system. Various features such as particulate service, dependency based on the contents, Connection between the system and the user, the set of rules for communication, application of the developed software etc.

2.2 Uses of CBSE in Software Development

Although, use of the object-oriented methods have promoted the use of software's and therefore there is an enormous gap between the entire systems and is various categories. In order to fill overcome this gap, several attention-grabbing concepts have evolved in object-oriented software for the recent years mentioned in[5,8,9]. CBSE has been generally utilized in the development of software, because it enhances reusability and adaptability, therefore it reduces the prices and risks which are concerned in the development of this systems[10–12].

3. Software Development based on Component Systems

In general, the CBSE proposes that the software development based on component systems varies from the standard model of development. Major characteristics of ancient package development and the component-based package development are shortly mentioned with in the following sections.

3.1 Component based Design

The diagrammatic representation of the general approach is shown in Figure 1 for developing a system and enhances the functions incrementally by adding and/or replacing the elements present in it. In order to obtain such potential architecture, a novel design of software system package is more important. Recent component-based systems use the novel package design like the Microsoft Foundation Class. They are provided within the type of frameworks. Frameworks are practicable for the underlying package design.

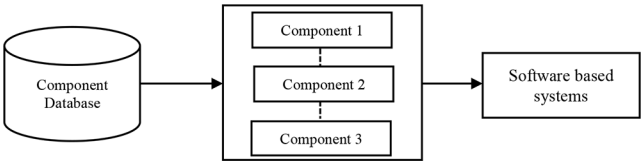

Figure 1. General approach of CBSE.

3.2 General Approach

In this general approach, making a package, its deployment and delivery are done as a step by step process. Since some elements of a system are non-heritable from the software developers and/or be done as outsource for an alternative in the various organizations, some elements of package methods are done at the same time.

4. Projected Methodology

An outline of the proposed methodology is depicted in this section. This method has to cope with the choice of the component from the repository, development of the component, integration of component, regression of component and composition of the element. Component-based software is obtained as a result of the composition of some components with the interfaces outlined and it is shown in. In CBSE, choice and combination of these components need their interface while not displaying the choices. This technique is extremely just like the ideas of Object-oriented ways. However, the component-oriented approach specialized in inheritance instead of reusability. Combination of these components desires an outlined and customary component of multiple options and is mentioned[2].

4.1 Identification and Design of Components

The identification and design of components consists of the following factors: choice of the component, Prioritization of the component, and Customization of the components.

4.1.1 Choosing the Component

As per the appliance domain and gathered necessities, each and every component should be hand-picked from the components database. This choice could embrace the availability of precise components and checking the availability of the changeable components.

4.1.1.1 Availability of Precise Components

Components are the elements that can be used to fulfil the mark necessities of a software development process. These components can be used since it cannot be modified. The component which is used should be potential such that one element can be easily accessible and also it should be consistent with the application and current research. The components are chosen due to these factors.

4.1.1.2 Checking the Availability of Changeable Components

These elements are same because the categories were derived from the Object oriented Software's; where it have a tendency to don't must begin from the scratch instead we will acquire associate accessible element and create changes based on the users demand.

4.1.2 Categorization of Software Components

Categorization of the software Components isn't the only criterion to spot the elements. In this identification process, the users will rank the elements consistent with the applications demand and desires of the user. Element categorization could facilitate once we need to deliver our software packages in increments. We will rank our necessities and so as to deliver most simple practical package, we have a tendency to rank our elements. Categorization of software Components could include the following factors: Consistent Components and Accessible Components.

4.1.2.1 Consistent Components

The software developers can rank the elements consistent with their necessities. They will choose those elements which can facilitate in delivery of the initial software package. These elements are hand-picked when there exists a consistent of necessities. The most important necessity of a component is it can be delivered during the initial process and then if there exists any increment process, it can also be carried out. Consistent Components could include the following factors: Accessible Components, Accessible Components and Primarily based Components.

4.1.2.2 Accessible Components

Second criterion is the supply of the accessible elements towards the database. It will deliver the package supported by simply accessible elements, in order to initiate the fast delivery of the package. It will facilitate in quicker integration as well as fast deployment of the package. When integrating and delivering the accessible elements, the software developers will choose modifiable elements and for faster deployment of latest elements.

4.1.2.3 Primarily based Components

This criterion is supported by the categorization of elements that are fulfilling the overall structure of the package. It should embrace either choice of obtainable element or the deployment of latest software.

4.1.3 Customization of Components

Customization of components is the method of addressing the problems associated with the choice and preparation of optimum elements. The components have a property that it can't integrate an excessive amount of programmed, since it can ultimately increase the complexity of the software. The Choice of choosing components should be supported by the following factors:

1. Components should be distinctive to provide Same Functionalities
2. Combination of elements to provide various connected options
3. Removal of Featured elements which are ambiguous in nature
4. Removal of Extraneous Interactions among elements

5. Conclusion and Future Enhancements

By the use of the component-based systems in software engineering, modification for various means of software package development can be done. Applications of the component-based systems in the Software package development is considered as the standard method, standard design and specialised associations that the developer can accumulate our technology and experience. Component based software engineering is an elementary technology for the process of software package development. It only needs the process of re-thinking of a spread of methods in software development. Even though these technical problems are there, various non-technical problems like commerce of elements and management problems are exists in the software package. Feature enhancements in this can be a new technology that could overcome the drawbacks present in this.

6. References

1. Szyperski C. Component software. Addison-Wesley.1998; 15(5):37–46.
2. Aoyama M. Component ware: Building Applications with Software Components. Journal of Information Processing Society of Japan. 1996; 37(1):71–9.
3. Nautiyal L, Tiwari UK, Sushil C, Dimri SC, Bahuguna S. Elite: A new component-based software development model. International Journal of Computer Technology and Applications. 2012; 3(1):119–24.
4. D'Souza DF, Wills DC. Objects, components and frameworks with UML: The catalysis approach. Addison Wesley; 1998. p. 1–816.
5. Fayad ME, Schmidt DC. Object-oriented application frameworks. Communications of the ACM. 1997; 40(10):32–8. https://doi.org/10.1145/262793.262798
6. Ning JQ. A component-based software development model. Proceedings of 20th International Computer Software and Applications Conference: COMPSAC '96; 1996. p. 389–94. https://doi.org/10.1109/CMPSAC.1996.544597
7. Enterprise JavaBeans: Server Component Model for Java [Internet]. [cited 1997]. Available from: http://www.javasoft.com/products/ejb/.
8. Sommerville J. Software Engineering (6th Edition). Addison-Wesley; 2001. p. 42–56.
9. Vidger M. The evolution, maintenance and management of component based systems. Boston: Addison-Wesley; 2001. p. 527–39.
10. Patel S, Kaur J. A study of component-based software system metrics. International Conference on Computing, Communication and Automation (ICCCA); 2017. p. 1–310.
11. Basili VR, Boehm B. COTS-based systems top 10 lists. IEEE Computer. 2001; 34(5):91–5. https://doi.org/10.1109/2.920618
12. Chen J. Complexity metrics for component-based software systems. International Journal of Digital Content Technology and its Applications. 2011; 5(3):235–44.