Software Reuse: Component-Based Development Issues

Hudaibia Khalid
Bahria University
Islamabad, Pakistan

Abstract:

Component-based development (CBD) supports modification and integration of reusable software components to develop complex software systems with minimum effort and cost. Reuse of software products is reducing cost of software development, increasing productivity, and saving time. In the context of reuse, software reuse is in the development phase and has not achieved its full potential. Literature survey is performed on benefits, major research contribution and issues/challenges of software reuse and Component-based development. This study aims to identify some serious issues of CBD that developers facing and they need to consider. This study would be beneficial for developers who are interested in Component-based reuse so that they can correctly and easily use a component without any problem.

Introduction:

Software reuse is the process of using the existing software to create a new software system. Purpose of software reuse is to improve software quality and productivity. People interested in software reuse because they want to build systems that are large, reliable, less expensive, more complex and delivered on time. Almost all reuse approaches help software developers compare, locate, and select reusable software artifacts. Software reuse is an area in software engineering that helps developers to avoid developing software from scratch. Through this perspective, Component-based development (CBD) is the right approach. In component-based development (CBD) software systems are construct by collect components that has been already developed and prepared for integration.

Component-based development (CBD) has received significant attention among software developers, vendors and IT organizations. Most recent trends in software engineering show that future developments will follow the component-based development (CBD) path [1]. Distributed and web-based systems, desktop and graphical applications are examples of domains in which component-based approach has been very successful [10].Although CBD promises to improve the quality, software development processes, productivity and reuse. This approach reducing maintenance costs and increasing reliability, while supports complex and usually distributed applications. Good repositories must be available to make component-based development cost-effective and components to easily found.

Types of component-based reuse:

There are two types of component-based reuse:

- With Change
- Without Change

With Change:

Reuse with change means that using a component that requires changes and then the modified component is used. It is difficult and time-consuming. It takes efforts to determine which parts of the component that require changes.

Without Change:

Reuse without change simply means selecting a component from a software component database, and use it into new software being developed. The cost of developing the new software is zero.

Component-based development (CBD) provides many benefits, but it has also several issues that developers need to consider.

There are many advantages of using the Component based development (CBD) approach:

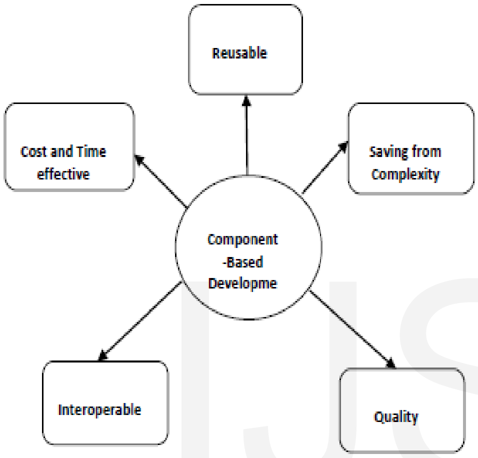

Figure 1: Advantages of CBD

Advantages	Description
Reusable	Reusability helps us to add more complex functionalities in our applications rather than concentrate on developing basic components.
Cost and Time effective	It helps to reuse the components again and again in the same applications that reduce time and cost.
Quality	The quality of the application developed from reusable components is also considerably improved.
Interoperable	It allows the components to intercommunicate with each other and this makes them interoperable.
Saving from complexity	It saves the programmers from the complex programming.

Table 1: Advantages of CBD

There are many issues of Component based development (CBD) in terms of reuse that developers need to consider:

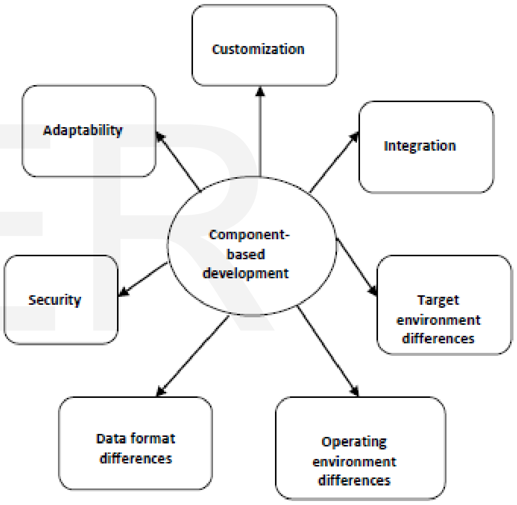
Customization
Adaptability
Integration
Component-based development
Security
Target environment differences
Data format differences
Operating environment differences

Figure 2: Issues of CBD

- **Customization**

It is a major issue in CBD. It simply means according to the requirements of new application customization of an already developed component.

- **Adaptability**

To use a component to a new platform if it were not developed for that platform is a main problem faced by many developers [2].

- **Integration**

The developers also face a major problem when the integration of a reuse component into new component.

- **Security**

Security is also a major issue for the developers who reuse the components available over the Internet. There may be a virus inside that component and may pass all the information of the business organization to attacker, who uses such an application [2].

- **Data format differences**

Software components that deal with data formats and encoding schemes cannot be reused without change in software.

- **Operating environment differences**

Software designed for slow input and output of data cannot be reused to support an environment where the input and output rate is very high.

- **Target environment differences**

Software components developed for one specific target environment cannot be reused without change.

Problem Statement:

Software reuse is the process of using the existing software to create a new software system. Component-based development (CBD) provides many benefits, but it has also several issues that developers need to consider. In this research the identification of issues which have been faced by developers that reuse a component.

Literature Review:

Current research shows that component-based software reuse is still facing different issues. The purpose of this literature review is to study, analyze and identify these issues regarding the software reuse and component-based software reuse. A comprehensive literature review has been done to obtain background knowledge on the topic. Following are the selected readings have been carried out to be more focusing towards the topic.

A) Major Research Contribution

Keswani, Joshi, Jatain summarizes [7] software reuse research and described major research contributions. By software reusing they make easier the development of software product by reusing the components of another software product in a different behavior. Software reuse is a simple idea of building and using software preferred parts. Building systems, pre-tested components will save the cost of designing, writing and testing. There is misunderstanding about how to implement and gain benefit from software reuse because practice of reuse has not proven to be easy. Software reuse is a difficult task especially for legacy software and it improves the quality and productivity of a software product.

B. Frakes, Kang summarizes software reuse research, major research contributions and unsolved problems. They discuss pointers to key publications, and introduce four papers selected from The Eighth International Conference on Software Reuse (ICSR8) [3]. Although significant progress has been made on software reuse and domain engineering, many important issues (scalability, sustainability, safety and reliability etc) remain. There is still much to do in reuse research before the vision of better system building through reuse.

B) Issues/Challenges

Kim summarizes difficulties in understanding the benefits of component-based software reuse. It also describes [4] before trying software reuse on a wide scale in a formalized way it must meet with key pre-conditions. Pre-conditions include proper training of all software developers and testers, including documentation, thorough programming guidelines, testing and programming, software design based on all key architectural factors in the development of software.

Gonzalez, Torres discusses the existent state of CBD and some of its critical issues for success. At first, they define [1] the difference between components and objects and how they are more suitable for software reuse, through flexibility and reliability. It is necessary to define the level and characteristics of specification and also to select an adequate specification language.

Crnkovic, Larsson described [8] the issues and challenges when developing and using an evolving component-based software system. The reuse provides many advantages, but it also requires systematic approach in design planning, extensive development, and in general more attention being given to components. They presented a case study as a successful example of the development of a component-based system. It is essential to have a proper strategy for migrating from old components to the new ones.

Gill discusses CBD and its associated challenges, and later outlines the issues of reusing component and its benefits in terms of cost and time-savings. Software reusability refers to reuse of a software component. Reuse software not only improves productivity but also has a positive impact on the maintainability and quality of software products. Considering the important issues related to software reusability, some guidelines related to reusability have been suggested, which will further help in enhancing quality and productivity activities within organizations adopting component-based development (CBD) [5].

Shiva, Shala wrote about [9] that software reuse has not achieved its full potential. Their research is focus on two major areas: enforcing reuse discipline in organizations, and component retrieval systems. Although, developing tools to easy the reuse process is not sufficient they also require tools to help managers impose reuse in organizations. Some organizations are adopting reuse successfully; but there is no standard of systematic reuse.

C) Benefits

Qureshi, Hayat presents different architectures of CBD such as ActiveX, common object request broker architecture (CORBA), remote method invocation (RMI) and simple object access protocol (SOAP) [2]. The overall objective of paper is to support the practice of component-based development (CBD) by comparing its advantages and disadvantages. Results show that component-based development (CBD) is cost effective, saves time and productive for the software development community.

Cybulski described in their paper the concepts of software reuse, reusability, reuse artifact and reuse library. It mention only those attributes of software artifacts' which have being reused, e.g. they have to be expressive, transferable, additive, formal, machine representable, self-contained, language independent, able to represent procedures and data, simple and easily changeable [6]. Finally the paper describes the benefits and the disadvantages of software reusability, focusing on the myths and misunderstanding held by developers and their managers.

The literature review discussed above provided information about the software reuse, its benefits, challenges and its contribution. As well as they also described its importance. Somehow some organizations are adopting reuse successfully but there are some misconceptions about software reuse and there are some issues in adopting software reuse.

Conclusion:

Software reuse increases software productivity. This paper highlight the issues related to reusing component-based development. Component-based reuse is not an easy task. It requires deep research and analysis.

Literature survey is performed on benefits, major research contribution and issues/challenges of software reuse and Component-based development.

Future work should be focus on solving all these issues and limitations, which will help in enhancing quality and productivity activities within organizations adopting CBD.

References:

[1] Rafael Gonzalez, Miguel Torres, "Critical Issues in Component-Based Development", January 2006.

[2] M. Rizwan Jameel Qureshi, Shaukat Ali Hayat," The artifacts of component-based development", ISSN 1013-5316, CODEN: SINTE 8 Sci.Int. (Lahore), 19(3), 179-185, 2007 188.

[3] William B. Frakes and Kyo Kang," Software Reuse Research: Status and Future", IEEE transactions on software engineering, vol. 31, no. 7, July 2005.

[4] Won Kim," On Issues with Component-Based Software Reuse", Journal of Object Technology, vol. 4, no. 7, September-October 2005.

[5] Nasib S. Gill," Reusability Issues in Component-Based Development", ACM SIGSOFT Software Engineering Volume 28 Issue 4, July 2003.

[6] Jacob L. Cybulski," Introduction to Software Reuse", Technical Report TR 96/4.

[7] Raman Keswani, Salil Joshi, Aman Jatain,” Software Reuse in Practice”, 2014 Fourth International Conference on Advanced Computing & Communication Technologies.

[8] Ivica Crnkovic, Magnus Larsson,” Challenges of component-based development”, The Journal of Systems and Software 61 (2002) 201–212.

[9] Sajjan G. Shiva, Lubna Abou Shala,” Software Reuse: Research and Practice”, International Conference on Information Technology (ITNG'07).

[10] Ivica Crnkovic, Stig Larsson, Michel Chaudron,”Component-based Development process and Component Lifecycle”, Software Engineering Advances, International Conference on Oct. 2006.