

Role of Component Certification in CBSE Activities for Building High Quality Software

Lina khalid Ahmed
Department of Software Engineering
Zarqa University
Amman, Jordan

Abstract— one of the main goals of software engineering is to build a software product of a high quality and then improve this quality. CBSE (Component Based Software Engineering) is the most important approach to build such type of product and that is because CBSE depends on the ability to choose the most appropriate components in order to use them in a specific application. The only way to choose the precise component is through its certifications. This paper defines the component certification and its role in the CBSE and then describes how this certification is used for building a high quality software product.

Keywords- Component; Component interfaces; CBSE approach; Component certification; CQM.

I. INTRODUCTION

Development with reuse has become an approach for building systems. It has been used in response to the demand for increasing time to market, lowering software cost, and producing a high quality software product (or product).

When a reuse concept is used, the process includes an activity where the abstract concepts begin to create executable reusable components.

CBSE is one of the important approaches for developing with reuse and it becomes one of the important approaches for building a high quality product. The important thing that makes this approach different is the existing of certification that is associated with each component. Component certification provides correct and complete information about the component; it provides functional and non-functional properties.

This paper introduces the effectiveness of component certification through CBSE activities and its position. It also shows how this certification helps the developers integrate trustworthy components and shows how it can build a high quality product.

This paper is classified as follows: Section 2 describes the related works and includes all the authors who have worked in this area. Section 3 defines software quality. Section 4 describes the CBSE. Section 5 describes the result. Section 6 sums up everything in the conclusion.

II. RELATED WORKS

Many researchers work on component technology that is based on reusability. [2], the authors propose a concept called SCL (Software Certification Laboratories); they suggested that this concept must take part in the certification product role which then offers trust to the consumer. In this method, SCL took all the information from the developer's site and passed it to the consumer's site and then returned back to collect this information from the user site and used it to produce the warranties according to these results. Another work is [3], in which the goal of this work was to describe a method to measure and certify the ability of software component to perform the reliability. The result of this technique is possibly certified components with the system that contains the components in order to define a component quality model, determine the evaluation techniques that will be used to evaluate the properties provided by the component quality model and to build a certification process to evaluate and certify software components. In [4], the authors in this paper describe the relationship between the evaluations performed during certification and their selection. They proposed a components-based life cycle for COTS (commercial-off-the-shelf) and software product line development. Moreover, they identify the process characteristics between the two types of evaluation and finally classify the required qualities based on their appropriateness to be evaluated during certification and/or during the development of system. In [5], an entire component-based business document modeling was produced. This modeling is built upon existing standards that are extended by introducing the concept of generic business document template out of the specific needs of the user's document. The result part of this paper is a complete library of reusable business

components that have been developed to easily produce a new business model for the system. [6], in this paper authors presented a new approach which is called CBQM (Component Based Quality Model) and discussed some issues which directly or indirectly affected the quality of the model.

This paper describes the importance of component certification through CBSE activities. It describes the major activities of component based approach then defines the importance of this certification to build a high quality product.

I. SOFTWARE QUALITY

There are many different definitions for the term quality. According to ISO/IEC9001, the definition of the quality is the capability of a software product to confirm to requirements [7], so software quality is the study of desirable attributes for the software product.

A quality attribute (QA) is a considerable property of the system that is used to show how the system will satisfy the needs of the stakeholders, so the quality attributes indicate the goodness of the software product along to some aspects of concern to a stakeholder [8].

The fundamentals of quality management are used to improve the quality of the products that are being made.

Software quality management for software systems has three main concerns [1]:

- At the organizational level, quality management is concerned with creating a framework of the organizational process. This means that the team of the quality management should take into account the responsibility of choosing the most appropriate development process for the system.
- At the project level, the quality management is concerned with the quality plan for a project. The quality plan should define the quality goals for the project and specify what standards and process are to be used.
- Quality management is also connected to the application of the specific quality process, checking that these planned processes have been followed and ensuring that the results are confirmed with the standards that are applicable to the project.

II. COMPONENT BASED SOFTWARE ENGINEERING (CBSE)

The objective of this section is to describe the essential terms that are used in this type of approach and then to show how this approach works through its activity.

A. Component and component model definitions

In the software industry, the term "component" is used in many different ways. For example it is applied to user interface components implemented as Active X, to major infrastructure items such as database management system, and to any software artifact that can be reused.

We can define a software component as a separable piece of executable software that can interoperate with other components within supporting environment and it is accessible only through its interfaces. In order to combine with other components, it must be able to achieve details of its interfaces [9]. Defining components as separable entities makes it possible to achieve the maintainability benefits offered by components. It also makes the reuse of components easy in specific scenarios, such as requirements for a new user interface, or changes of the structure of the company [9].

Some essential characteristics of components that are used in CBSE approach are [1]:

- Standardized: which means that the component used in a CBSE activities has to match with some standardized component model (which is defined later in this section)
- Deployable: the component must work as a stand alone entity on a component platform that implements the component model
- Documented: components should be fully documented so that the user can make a decision whether the selected component meets its needs or not

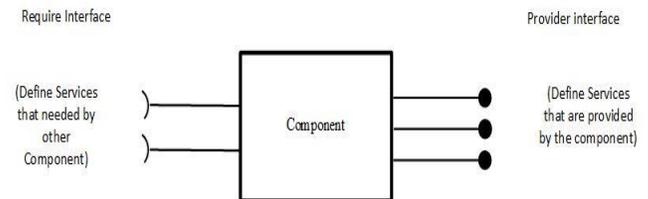


Figure 1. The component interfaces

As figure 1 shows, the components have two related interfaces. These interfaces reflect the services that the component provides (provider interfaces) and the services that the component requires (require interfaces) in order to work properly [1].

A component model can be defined as a standard for component implementation, documentation and deployment. These standards are for component developers and providers; for developers to make sure that the components can interoperate, and for the providers of the components who provide middleware to support component operation.

B. CBSE approach

CBSE is a process that highlights the design and construction of systems using reusable software components. By that definition, we can assemble components from a catalogue of reusable components in a cost and time effective manner [10].

CBSE has become the most important approach because all software systems became more complex and larger than before.

The only way to deal with this complexity was through reuse rather than re implement. The essential activities for CBSE approach with reuse are [1]:

- **Component analysis:** the stakeholder of the system defined their requirements in the abstract view rather than in details. Here the complete set of requirements is defined in order to identify the complete set of components. Usually there is no exact match between the selected component and the requirements, but the selected components may be used only to provide some functionality of the requirements
- **Modify requirements:** requirements are modified and refined according to the available components. If the user is not satisfied from the available requirements, he can switch to other requirements that support his request at this stage. If the modification is impossible, the first stage is repeated to search for other components for an alternative solution
- **System design with reuse:** the architectural design of the system is built. Further refinement for the design and more searches for the component are made through this stage. In this phase, the designer takes into account the components that are reused and organizes the framework to accommodate with them
- **Composition process:** the discovered components are integrated with the component model infrastructure. So the integration process is part of the of the development process. Figure 2 represents the main activities of CBSE approach.

Figure 2 shows that there are two additional stages, one at the beginning (requirement specification), and one at the end (system validation). They are not defined in the previous stages because they are the same in all other approaches so what is mentioned is only what makes the CBSE approach different, and that is what is demonstrated in the gray ellipses in the figure.

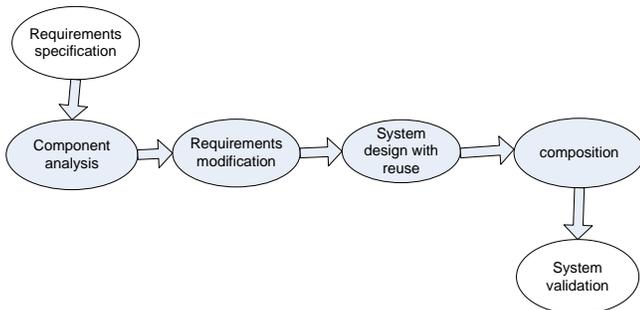


Figure 2. The Basic Activities of CBSE

C. Component certification

Component certification is a process that ensures that the software components match to basic standards. Based on this certification, trusted components are integrated. Moreover, the certification stands out as an important area to evaluate the

component reliability level, although that task seems to be very difficult because the community of software engineering has expressed many propertities to evaluate the specific components. [3, 12] a certification body must have a certification system that describes techniques for performing the certification process. A certification process can be defined as a process of verifying properties which are related to a component and this certification can provide the validity to that component [11]. The most important stages in the software component certification process are shown in figure 3:

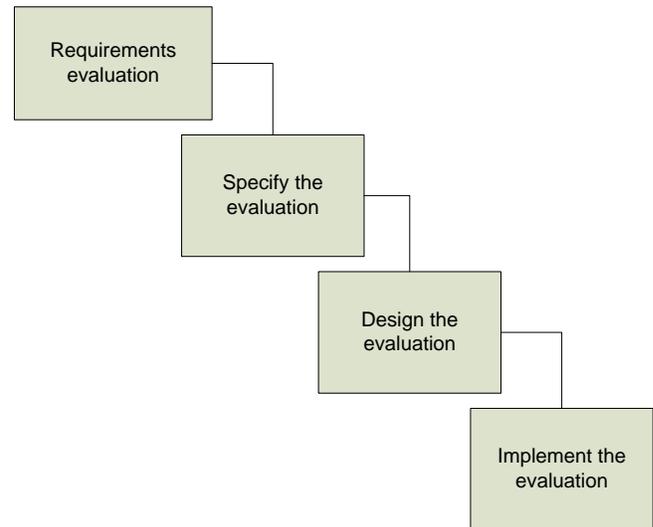


Figure 3. The component certification process

III. RESULT

Component certification and quality

High quality products that are based on CBSE approach always need an effective certification. Some issues need to be taken by organizations to make CBSE successful. These issues are:

- **Analyzing the domain** which identifies and distributes a set of software components that can be used in existing and future software in a particular application domain. The overall goal of this issue is to enable software engineers to share these components
- **Component qualification and composition:** they ensure that the chosen component is properly fit to the architectural style that is proposed. After that, when the component is qualified for reuse within an application architecture, conflicts may occur so the adaptation process technique (called component wrapping) is used. Component wrapping customizes the components to remove the conflicts through integration. Component composition assembles all adapted and engineered components to distribute the suitable architecture for an application
- **Analysis and design for reuse**
The requirement model is first analyzed and the elements of this model are compared to the

descriptions of the reusable components. If the components match the requirement, it will be reused in the system from the library of reusable components which is called a repository of components. Otherwise the component must be created to fit the requirements in the model and this should create reusable components.

- Classifying and retrieving components
A 3 C model (concept, content and context) is described to reuse an appropriate component in any system. The concept is a description of what a component does and the content of the component describes how the concept is achieved. The context places a reusable software component within its domain application.

The result of these issues must be certified components that can be stored in a repository of components to be used by the developer through applying CBSE approach. The developer looks for the appropriate components, modifies them as needed and incorporates them in the system. Selecting the most appropriate components makes the CBSE approach have good advantages such as reducing the amount of software to be developed and also reducing the cost and time, all of which leads to a faster delivery time to market.

The position of component certification is explained in figure 4.

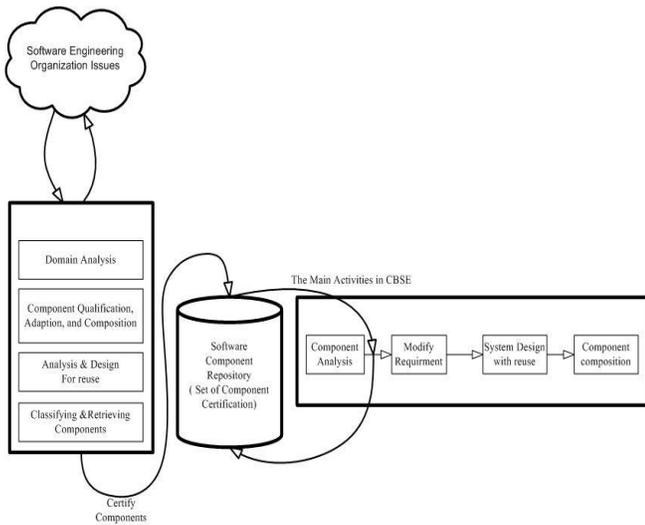


Figure 4. The position of component certification in CBSE

The role of the component certification appears through the first and second stages of CBSE activities. The requirements are specified and the right component is chosen. That is done according to some considerations that are specified through component certification which is stored in the repository of components.

The most important criteria that should be taken into account through selecting the components is the quality of the product.

The quality attributes of the product is very important through selecting components. The final architecture of the

system must provide the key attributes of the system, so the components that are selected and assembled must provide those quality attributes.

Component Quality Mangment (CQM)

Now after we see the position of component certification in CBSE activities we are going to use this certification to build a high quality product. Figure 5 shows the framework of the quality model for any component that is what we call Component Quality Model (CQM).

High quality products are always achieved through the integration of high quality components so the metrics are used through a process and then through a product as shown in figure 6.

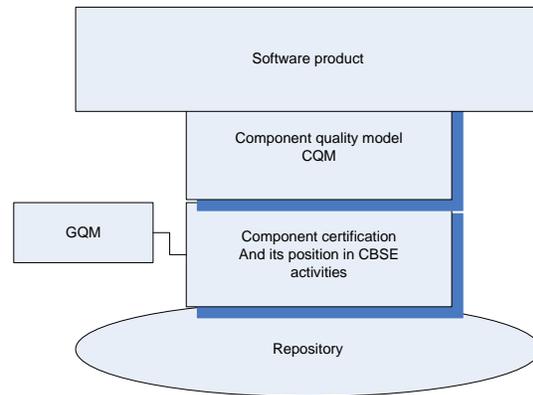


Figure 5. Component Quality Model (CQM)

GQM (Goal Question Metric) is an approach to decide what measurements should be taken and how they should be used. It is applied to process improvement which separates organizational concern (goals) from specific concern (the questions).

Metrics in this approach are the measurements that need to be collected to help answer questions and to confirm whether process improvements have achieved the desired goal.

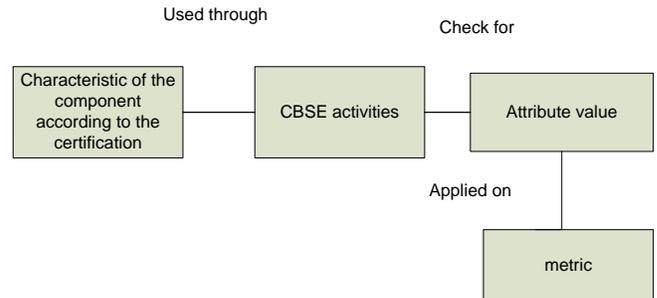


Figure 6. Quality attribute process

There are two types of metrics: control metrics and predictor metrics. Control metric is applied on the components in a CBSE activity, such as the average time and effort that are required to repair the reported defects in order to achieve high reliable components. The predictor metric is associated with a

software product. It must be known that is impossible to measure the specific characteristics of the component directly. The basic characteristics such as maintainability, usability, reliability, etc. are called external characteristics that relate how the user sees the product. This is why each characteristic has an internal attribute, which is what the measure is going to be applied on. For example: reliability (external characteristic) is associated with cyclometer complexity, size of the program and number of error messages (as an internal quality attributes).

IV. CONCLUSION

The idea of using CBSE approach is to build a system from existing components by assembling and integrating them. What makes this approach very useful is the existence of the certification. Component certification is a very important part in this approach because it ensures selecting the right component for reuse. All in all, this paper concentrates on the component certification and its affect on the quality of the product, it describes that all components that are stored in the repository are of high quality which is represented by the component certification. That is done by the life cycle by the CQM which is the core of this paper.

ACKNOWLEDGMENTS

This research is funded by the Deanship of Research and Graduate Studies in Zarqa University /Jordan.

REFERENCES

- [1] Somerville I, "Software Engineering," Pearson Education, 2011. ISBN10: 0-13-705346-0.
- [2] T. Janner et al., "A Core Component Based Modeling Approach for Achieving E-business Semantic Interoperability," Journal of theoretical & applied E-commerce research, Vol 31, issue 3, PP: 1-16, 2008. DOI: 10.4067/S0718-18762008000200002.
- [3] Alvaro A., Almeida E., Meira S., "Software f Component Certification : A survey ,conference on software engineering and advanced applications,IEEE,2005.
- [4] Land R., Alvaro A., Crnkovic I., "Towards Efficient Software Component Evaluation: An examination of component selection and certification, IEEE computer society, 34th Euromicro Conference Software Engineering and Advanced Applications, 2008.
- [5] Tamal S. , Rajib M., "State-Model-Based Regression Test Reduction for Component-Based Software," ISRN Software Engineering, vol. 2012, Article ID 561502, 9 pages, 2012. DOI:10.5402/2012/561502.
- [6] Dixit A., " CBQM: Component Based Quality Model, "school of computing science and engineering, Galgotias University, August ,2013.
- [7] International Organization for Standardization, "ISO/IEC 24765: Systems and software engineering — Vocabulary," 2010.
- [8] Bass I.,Clements P., Kazman R.,"Software Architecture in practice", 3rd Edition SEI Series in Software Engineering , October, 2012.
- [9] Whitehead K., "Component-based Development: Principles and Planning for Business Systems," Pearson Education, 2002.ISBN 0-201-67528-5.
- [10] Pressman R.,"Software Engineering: A practitioner's approach;"McGraw-Hill, 2010.
- [11] Stafford J., Wallnau K. , "Is the Party Certification Necessary?," Software Engineering Institute, Carnegie Mellon University,USA.<http://niap.nist.gov/howabout.html>.
- [12] Alvaro A., Land R., Alvaro A., Crnkovic I.,"Software Component Evaluation: A theoretical study on component selection and certification, "Malardalen university, 2007, ISSN 14404-3041.