A Survey on Component-Based Software Development System

Gaurav Sethi[1], Dr. Andhe Dharani [2], Ashwini V. Patil[3]

[123]R.V. College of Engineering, Bangalore

***Abstract* - The primary goal of component-based software engineering is to address the development of systems as an erection of system components, and development component as the reusable entities and maintenance or upgrading of system. Component-based software development (CBSD) is an transpire rule that promises to take software engineering into a new epoch. The CBSD methodologies dealing with COTS components. Component based software development is integrated with a shift from statement-oriented coding to building a system by plugging together with components. The established disciplines from software engineering needs a new methodologies to support the component-based development. This requires traditional methodologies and tools to support the entire component and system's life cycle including organisational, marketing, technological and other aspects. The paper define relevant terms by relating the definitions to the already well-defined terms of object technology. The paper describe new vital activities in the component-based systems life cycle, and also discusses the technical and non-technical issues that need to be resolved for widespread assumption of this approach. This paper also presents several directions for future research and work in this area.**

***Keywords* - Component-based software development (CBSD), Component-based software Engineering (CBSE), COTS (commercial off-the-shelf), quality-of-service (QoS).**

I. INTRODUCTION

The CBSE represents the “the ‘buy, don’t build’ ideology". Because the huge uses of the components and the CBSE process is slightly different from the waterfall approach. It accent on developing of the new software from the pre-built components. The quickly stem demand for rapid and cost effective development of large-scale and complex software systems has introduced a major dispute for the software community.[1] The new era is to adopt the component-based software development approach to overcome the dispute. The component-based software development approach is based on the concept of developing software systems by choosing the building blocks of a new system from commercial off-the-shelf components and assembling that components with in a software architecture. Therefore, the component-based software development approach has great probable for reducing the development cost and time. To build a component-based software system, organization may select from COTS developed by internal or external source. The development of CBSD involves new vital activities such as customization, integration and evaluation COTS components.[2] The Model-driven engineering (MDE) techniques are increasingly being used many of development and operational life cycle which concerns with the large scale of component-based development systems. The lacking significant of research deals with the QoS property of component-based systems throughout their development life cycle.

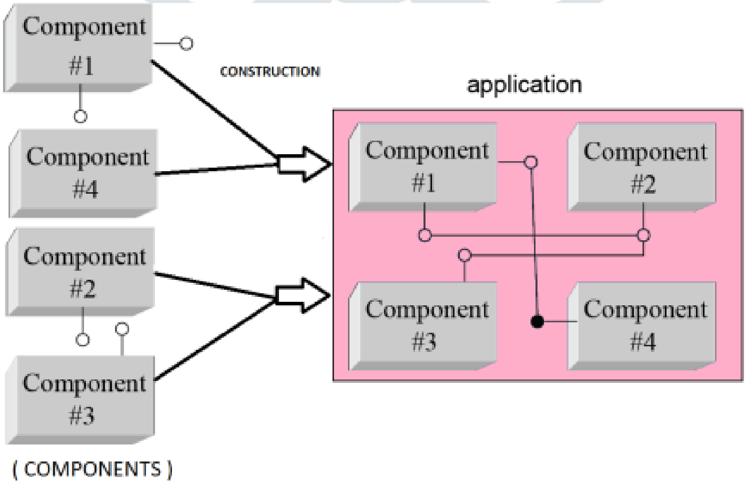

Figure 1 (Constructing CBSD)[3]

II. Frameworks in CBSD

1. **Application Framework:** The Application Frameworks are the sub-system design which is made up of a collection of classes and the interfaces between them. The classes may be abstract class or concrete class. The sub-system is imposed

by adding the components of design and by instantiating with the classes in framework. Frameworks are relatively large entities that can be reuse.

2. **Framework Classes -** System infrastructure framework: The system infrastructure framework is support the development of system infrastructures like user interfaces.
3. **Middle-Ware Integration Framework -** The middle-ware integration framework is the standard and classes which supports the information exchange.
4. **Enterprise Framework -** The Enterprise application framework is supports development of the specific type of application like financial system.
5. **Extending Framework -** The extending framework are universal and extended to create more specific sub-system or application. The extending framework includes creating the new concrete classes which inherit the operations from pre-existing abstract classes in framework.[5] and creating the methods that called in response to events which recognized by framework.

III. Component Software

The software components are unit of software designed which interact with the other independently developed components. Software component engineering is mainly focuses on the packaging of software into the independent units to allow maximum re-usability. The Component based software development (CBSD) is an adaptation of the object-oriented software development (OOD) and share the goal of software re-usability.[6] the object-oriented software development is an implementation methodology and the Component-based software development is an interface methodology. In Component-based software development the significance is on standardizing interface between the components, without restrictions on how to implementation accomplished. Therefore the Component-based software development is closely relate to the module design in the separate interface and implementation.

In the object-oriented software development, the code reuse is competent through the inheritance of implementation code. While object oriented languages also allows the separation of implementation and interface. The class libraries and the procedural libraries are designed to be assimilate into the larger applications are successful example of software reuse.

Life-Cycle of CBSE

In CBSE life cycle various phases are :

- Requirements analysis
- Software architecture selection, construction, analysis, and evaluation
- Component identification and customization
- System integration
- System testing
  - Functional or Black Box Testing,
  - Structural or White Box Testing
- System maintenance

Requirements analysis and specification - The services of systems, constraints and their aims are established (such as specification what the system to do). The Component requirement analysis is the process of discovering, documenting, understanding, validating & managing the requirement for a component. The objective of component requirement analysis to produce the complete, relevant, consistent and requirements that a component should realize.

Software architecture selection, construction, analysis, and evaluation - The Component development is process of develop and implement the requirements for a functional component with the multiple interface. The objective of component development System are the final component products and development document.

Component identification and customization - The objective of component identification is select and test the components and also check whether the components are satisfy the system requirement with reliability & high quality. The Component customization is processes that involve the modifying components for a specific requirement. and doing necessary change to execute the component on a platform. it is also upgrading the specific component to get a better performance or a higher quality.

Implementation and unit testing - The development of design in executable way, which composed of the smaller units. System testing is a process of evaluate the system to confirm that system satisfies the all specified requirements and also identifies the correct or defects in system implementation.

System Maintenance - The System maintenance is a process of provides the service and maintain the activities which is needed to use the software effectively after it delivered to the user.
The objective of system maintenance to provide the services to end-users while it correcting faults or improves the software performance or some other attributes of components.

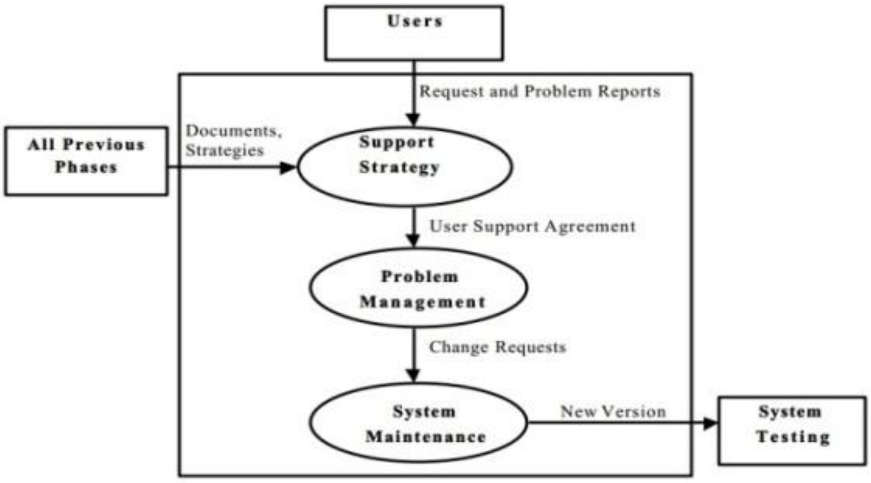

Figure-2 (System maintenance process)[9]

IV. Conclusion and Future Work

In this paper, the component-based software developing technique and cannot be fully deployed if the development organizations are not embraced according to the basic principles of the Component Based Software Engineering[8]. This requires adjustments of the development processes. This approach goals for the increased reusability of pre-developing components, the performance for the application decrease and the performance for verification increase.

References

[1] Roger Pressman, Software Engineering: A Practitioner's Approach, Fifth Edition

[2] Crnkovic, Larsson, Michel Chaudron Component-based Development Process and Component Lifecycle, ABB Corporate Research , Vasteras, Sweeden

[3] Jifeng, Xiaoshan Li, Zhiming Liu Component Based Software Engineering – The Need to Link Method and their Theories, UNU IIST Report No 330

[4] Martin Wirsing, A Component Based Approach to Adaptive User-centric Pervasive,13th International Symposium on Component Based Software Engineering, 2010

[5] Stephen Thesing, The Joy of Qualifying Siftware in the Avionics Area, 13thInternational Symposium on Component Based Software Engineering, 2010

[6] A.W.Brown, K.C. Wallnau, "The Current State of CBSE," IEEE Software, Volume: 15 5, Sept.-Oct. 1998, pp. 37 – 46.

[7] M. L. Griss, "Software Reuse Architecture, Process, and Organization for Business Success," Proceedings of the Eighth Israeli Conference on Computer Systems and Software Engineering, 1997, pp. 86-98.

[8] P.Herzum, O.Slims, "Business Component Factory - A Comprehensive Overview of Component-Based Development for the Enterprise," OMG Press, 2000.

[9] Component-Based Software Engineering: Technologies, Development Frameworks, and Quality Assurance Schemes Xia Cai, Michael R. Lyu, Kam-Fai Wong Roy Ko The Chinese University of Hong Kong Hong Kong Productivity Council.