# An Analysis Of Life Cycle Models For CBSD

Menal Dahiya
Department of Computer Science, MSI, Janakpuri, Delhi, India
menaldahiya@gmail.com

***Abstract*—The need for computer software development is fast growing in our society today. The major challenges faced by the industry is to provide products with quality and functionality at low cost and short time. To meet these challenges software development must be coped with complexity and to adapt quickly to changes. The solution of this is reusability. A number of software life cycle models existing today focused on building on traditional software systems. But the question is whether these existing models are ready to fit in this emerging field. Several life cycle models for component based software development have been introduced by researchers. The role of component based software engineering is to develop the reusable components and assemble them into one system. In this research, we surveyed some of the popular approaches and provide a comparative study of these approaches.**

**Keyword-** Component, Component Based Development, Life Cycle Model, Reusability.

## I. Introduction

In the present Era of Information Technology, software development community is continually seeking new technologies for enhancing the productivity and quality of software. Software systems have become larger and more complex than before. A lot of research has been done to reduce the development cost as well as development time. Following the traditional approach to find out the solution of the problems is not sufficient. So it has become the challenge for the researchers to find way to develop software in a fast and cost effective way. Every time if the software products are developed from the scratch, it takes a lot of time and money, hence the concept of software re-usability was developed which created the idea of component-based software engineering (CBSE). CBSE has substantially impacted software development in terms of both technical and organizational change of the process. Now CBSE has become a very popular and effective approach to software development. Component-based development has improved quality, increased productivity, effective management for complex software and wider range of usability [1, 2].

Although CBSE is a demanding field of software engineering and proven its importance in the area of software engineering, but it still needs attention of researchers to deal with a number of issues or challenges. Although component based software development (CBSD) is a popular area of software engineering, it is steadily growing towards its maturity. However, there is so little knowledge about life cycle models for CBSD. In this research, we surveyed on Several popular life cycle models and tried to provide a comparative study based on the strengths and weaknesses addressed by the researchers. The rest of the paper is as follows: Section II gives an overview of component-based software development. In Section III, the survey of different life cycle models is presented and final section IV, concludes the paper.

## II. Component Based Software Development

According to Sommerville, "CBSE is the process of defining, implementing and integrating or composing loosely coupled independent component into systems" [3]. The component has many properties like independency, standardization, interactive interfaces, reusability and give specified services.

### A. *Independency of Components*

Components need to be totally independent in such a way that if the components: Components need to be totally independent in such a way that if the component is removed or replaced, then the system need not be changed.

### B. *Standardization of Components*

Standard is an important factor for the integration of the components. Proper standardization can guarantee the use of components which are developed using different language.

### C. *Interactive Interfaces*

It works as the component integration support for distributed components to work together.

### D. *A Process of Development*

For the efficient reuse of components, a development process tailored for CBSE is required.

## III.LIFE CYCLE MODELS OF CBSD

Component based software systems are built from well-defined, independently produced self made components and assembling them together rather than programming on overall system from the scratch. Component based software models focus on reusing the whole software components Traditional life cycle models basically do not focus on artifact reuse. Thus the life cycle of the traditional software system is different from component based software system. A life cycle model is a blueprint of the software process. It describes all the phases in order in which the flow of process goes on. To develop a good software product, software engineers must identify a suitable life cycle model for their project. In Figure 1 Sommerville has provided a sequential approach for CBSD [3]. The approach comprises of the following Six Steps:

i. Step 1: The user requirements are developed in outline rather than in detail as specific requirements limit the number of components that might be used.
ii. Step 2: A complete outlined set of requirements is used to identify as many components as possible for reuse.
iii. Step 3: Requirements are refined and modified so that they can comply with components.
iv. Step 4: Architectural design is developed.
v. Step 5: After system architecture is designed, steps 2 and 3 may be repeated.
vi. Step 6: Finally the components are integrated which turns into a complete system.

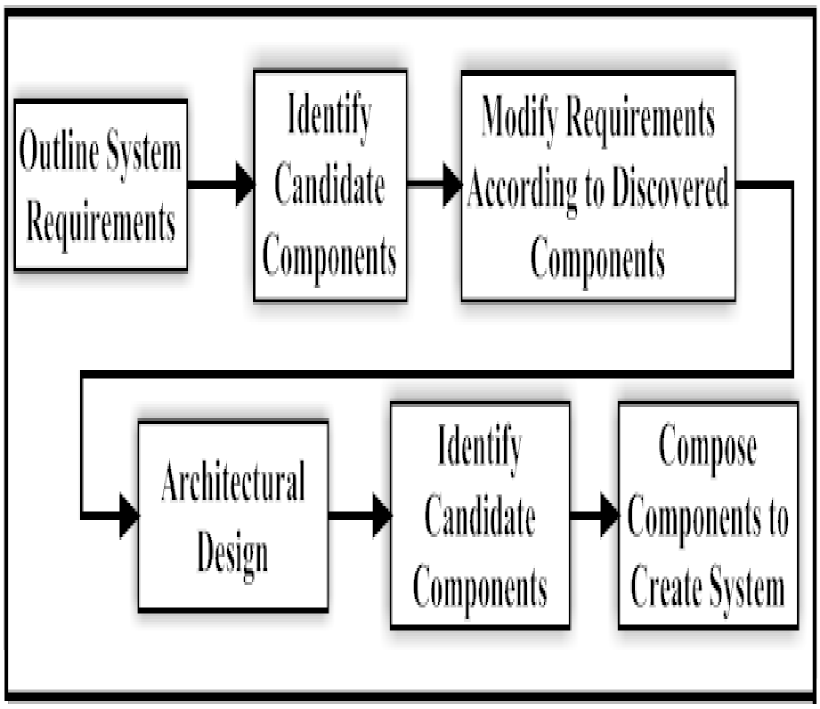
Outline System Requirements
Identify Candidate Components
Modify Requirements According to Discovered Components
Architectural Design
Identify Candidate Components
Compose Components to Create System

Figure 1. The CBSD Process.

There are various life cycle models for CBSD have been designed so far. In this section we will describe briefly some of those proposed models for CBSD.

### A. *The X Model*

In this X Model, the processes are started by requirement engineering and requirement specification. 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, it uses two main approaches, develop software component for reuse and software development with or without modification in reusable component [4].

### B. *The Y Model*

The Y Software Life Cycle Model describes software reusability during CBSD. The Y Shape of the model considers iteration and overlapping. Although the main phases may overlap each other and iteration is allowed, the planned phases are: domain engineering, frame working, assembly, archiving, system analysis, design, implementation, testing, deployment and maintenance [5].

The reusability within this life cycle is efficient and more effective than within the traditional models because it integrates at its core, the concern for reuse and the mechanisms to achieve it. The Y model supports “development with reuse” through component assembly, as well as “development for reuse” through component archiving. Y model shows great applicability in component based software development.

*C. The Y Model*

The V Model is a top-down approach to system design: the system is designed first and then components are developed. A straightforward adaptation of the V Model for CBD would be to retain the top-down approach to system design but uses a component as a module. However, such a straightforward adaptation of the V Model is at variance with the standard CBD process precisely because it does not include a component life cycle and consequently does not incorporate the bottom-up nature of CBD [6].

An adaptation of the V Model for CBD that does incorporate the bottom up nature of CBD is completed by containing separate life cycles for component development and system development. However, this adaptation really applies the V Model only to its system life cycle; there is no evidence of the V Model in its component life cycle. So, for adaptation of the V Model properly for CBD, both the component life cycle and the system life cycle are needed.

*D. The W Model*

The Two V Models are conjoined via the step of component selection, adaptation, and deployment. This ‘double V’ process is represented as the W Model. It consists of two phases: component design and component deployment, and is set in the context of a problem domain. In the design phase, components are designed and constructed according to the domain requirements or knowledge, and deposited into a repository [7]. Repository components are domain-specific but not system-specific. In the deployment phase, components are retrieved from the repository and instantiated into executable component instances which are then deployed into a specific system under construction. Bottom up approach is used for the process of component selection and adaptation, followed by system assembly, which is simply the composition of the deployed components. The bottom up nature of this process is indicated by an iterative loop. It is worth noting that within this loop, the component life cycle links up with the system life cycle, since deployed components are iteratively assembled into the system life cycle [8].

Applying the V Model to both the component and system life cycle yields a W model. The component life cycle of W model is similar to that in the Y model. However, the Y model does not apply the V model in any way to its component life cycle.

*E. The Knot Model*

A Component based model is described in which reusability in the form of the component is applied. The utmost emphasis is given on reusability, modularity, risk analysis and feedback in each phase, which results in simple, error free and a profound new system with proper feedback in each phase of software development. For a new CBSD, it uses all the three: the newer one, the existing component and the modified component to build up a new development. It consists of four phases. 1. Reusable Component Pool 2. Development of New Component 3. Modification of Existing Component 4. Development of New Component Based Software Development. The major advantages of this software life cycle is to handle the complex systems using modularity, reusability, lifetime process, evaluation & testing in each phase to reduce the risk [9].

This model emphasizes on reusability considering risk analysis and feedback on each and every phase. It is best suited for medium or larger complex system. A component that is developed is based on general specifications not on particular application’s specification. In this model risk is resolved in the early stages of each phase that results in the reduction of cost and time.

*F. The M Model*

The M model is a new Component based model which enhanced the concept of reusability in the form of the component. The M model represents a combination of Traditional models and component based software engineering model. The process steps start upward in a two way direction, on first phase, i.e. called development for reuse, after the storage of component in pool i.e. called second phase, the process steps moving down which is called the development from reuse i.e. third phase. The process steps are bent upwards in the fourth phase, which is called testing phase, and then it is once again moving down in the fifth phase to form the typical M shape. The M model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing. This model follows the both top down and bottom-up approach. At last after developing a system the component of the system place in the component storage pool for reuse in future [10]. M model is used for fast development and delivery of a high quality component based system at low investment cost. It improves the productivity of software.

The component process model can be a part of software development process. Table 1 shows the characteristics and limitations of the above stated models:

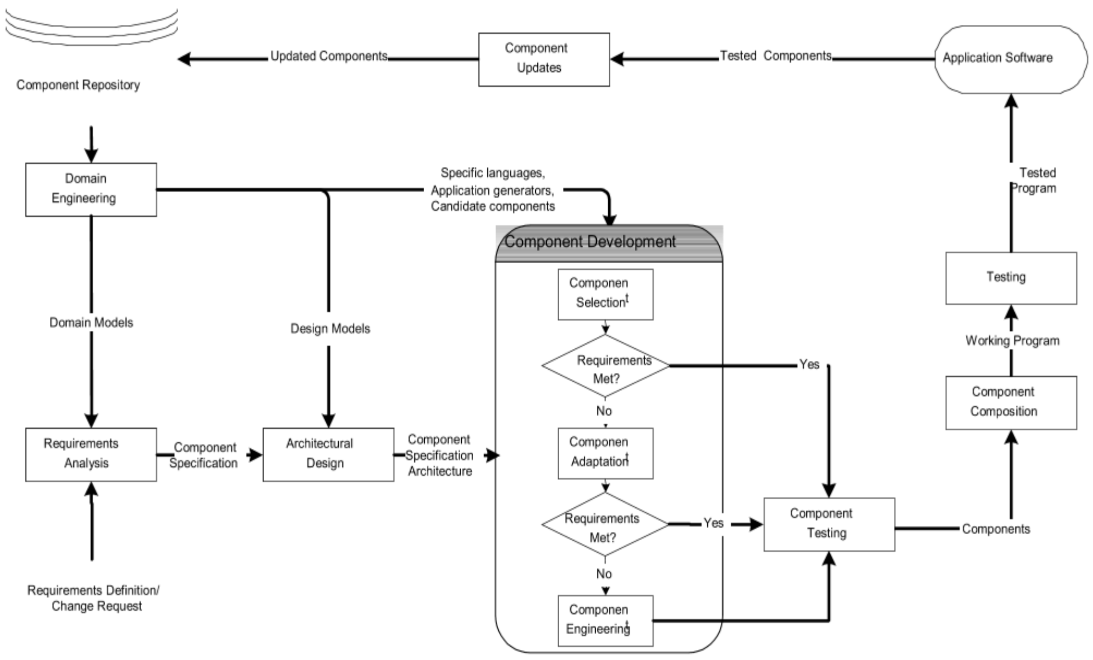

Figure 2. The MyCL Process Model

Table 1. Comparisons of CBSD Models

| CBSD Models | Characteristics | Limitations |
|---|---|---|
| X Model | Reusability, Clear Requirements, For Large Systems | Increases Complexity, No Risk Analysis, Increases Costs |
| Y Model | Reusability, Solving by Analogy, Follows both top Down and Bottom up Approach | Iteration and Overlapping during process |
| V Model | Reusability, Sequential Process, Top Down Approach, Verification and Validation Phase | No Tool Support, No Risk Analysis, Only Components are Reused |
| W Model | Reusability, separate life cycle for component and system, beneficial for practical system development | Heavy Process, Feedback is not allowed, High Risk, Iteration during the process |
| Knot Model | Reusability, Requirements clear, No Complexity of Software Applications, Reduces Risk and Development Time, Reduces Cost, Applicable to larger & Complex Systems | Selecting a right component is difficult, The reservoir may be huge or difficult to manage |
| M Model | Selection of Component from Storage Ppool, Best for Complex Systems, Reusability | Increases Dependability, Difficulty in removing Errors |

## IV. CONCLUSION

In this paper, we studied several life cycle processes for component-based software development and provided a comparative analysis of these processes. Component-based software engineering is still at its young stage of life and there is much room for research in this field. We noticed that the present life cycle models have several weaknesses. In our future research, we will try to propose CBSD life cycle model that can efficiently face the weaknesses of present CBSD life cycle models.

## REFERENCES

[1] Alan W. Brown, “Large Scale Component Based Development-First Edition,” Prentice Hall PTR, December 2000, ISBN: 978-0130887207.
[2] Ivica Crnkovic, "Component-Based Software Engineering -New Challenges in Software Development,” Proc. 25th International Conference on Information Technology Interfaces, 2003, pp. 9-18, DOI: 10.1109/ITI.2003.1225314.
[3] Ian Sommerville, “Software Engineering-Seventh Edition,” Pearson Addision Wesley, 2004, ISBN: 0321210263.
[4] N. S. Gill, P. Tomar, “X Model: A New Component-Based Model,” MR International Journal of Engineering and Technology, Volume. 01, Issue. 1&2, pp. 1-9, 2008.
[5] L. F Capretz, " Y: A New Component-Based Software Life Cycle Model,” Journal of Computer Science, Volume.01, Issue.01, pp. 76-82, 2005.
[6] Ivica Crnkovic, Michel Chaudron and Stig Larsson, “Component-Based Development Process and Component Lifecycle,” Journal of Computing and Information Technology, Volume.13, Issue.04, pp. 321-327, December 2005.

[7] Kung-Kiu Lau, Faris M. Taweel, "Domain-Specific Software Component Models," Proc. Of 12th International Symposium on Component-Based Software Engineering, pp. 19-35, 2009, DOI: 10.1007/978-3-642-02414-6_2.
[8] Kung-Kiu Lau, Faris M. Taweel, Cuong M. Tran, "The W Model for Component-Based Software Development," 37th EUROMICRO Conference on Software Engineering and Advanced Applications, pp. 47-50, 2011, DOI: 10.1109/SEAA.2011.17.
[9] Rajender Singh, Parveen Kajla, "A New Knot Model for Component Based Software Development," International Journal of Computer Science Issues, Volume.08, Issue.03(2), pp. 480-484, May 2011.
[10] M. Kaushik, "Reusability Concept Using an ''M'' Component-Based Model," International Journal of Science, Engineering and Technology (IJSETR), Volume.02, Issue.02, 2013.
[11] S. A. Fahmi, Ho-Jin Choi, "Life Cycles for Component Based Software Development," IEEE 8th International Conference on Computer and Information Technology Workshops, 8th -11th July 2008, DOI: 10.1109/CIT.2008.Workshops.82.
[12] Roger S. Pressman, "Software Engineering: A Practitioner's Approach-Fifth Edition," McGraw-Hill Higher Education, 2001, ISBN: 0072496681.

## AUTHOR PROFILE

Ms Menal Dahiya is Assistant Professor of Computer Science at Maharaja Surajmal Institute (Affiliated to GGSIP University, Delhi). She received her Ph.D from Maharshi Dayanand University, Rohtak, India in 2017 and MPhil in Computer Science from Chaudhary Devi Lal University, Sirsa, India in 2007.Before she had studied at Guru Jambheshwar University of Science & Technology (GJU), Hisar and KUK, Kurukshetra, India. Several of her research papers have been published in international peer-reviewed journals indexed in Scopus, ESCI, ICI and others.