

# W model of Component Based Software Development

Ishita Verma

*House No.4, Village Dayalpur  
Karawal Nagar Road  
Delhi-110094, India*

## ABSTRACT

Component based software engineering is an approach of software development that emphasizes the design and construction of software using reusable software components. A set of pre-built, standardized software components are made available to fit a specific architectural style for some application domain. The current models for component based software development do not take into account Verification and validation at different stages, properly. This paper studies a W model of software development using component based approach which exclusively includes verification and validation, like the V model of software development.

## General Terms

Component Based Software Engineering

## Keywords

Component based software engineering, W model, Component based development

## Abbreviations

Component based development (CBD), Verification & Validation (V&V)

## 1. Introduction

Component Based Software Engineering (CBSE) is an approach that relies on the reuse of Software components. A set of pre-built, standardized software components are made available to fit a specific architecture for some application domain. The application is then assembled using these components, rather than the discrete parts of a conventional programming language. Component based systems are easier to assemble and therefore less costly to build than systems constructed from discrete components. A Component is an independent executable entity. It does not have to be compiled before it is used with other components. The services offered by a component are made available through an interface and all the interactions take place through that interface. Advantages of Component based development are increased quality, accelerated development and reduced risk. The development process for component based software involves two separate processes: component development and (component based) software development.

The standard development process for component based systems does not include Verification and Validation. This paper studies W Model for CBD systems, which is like V

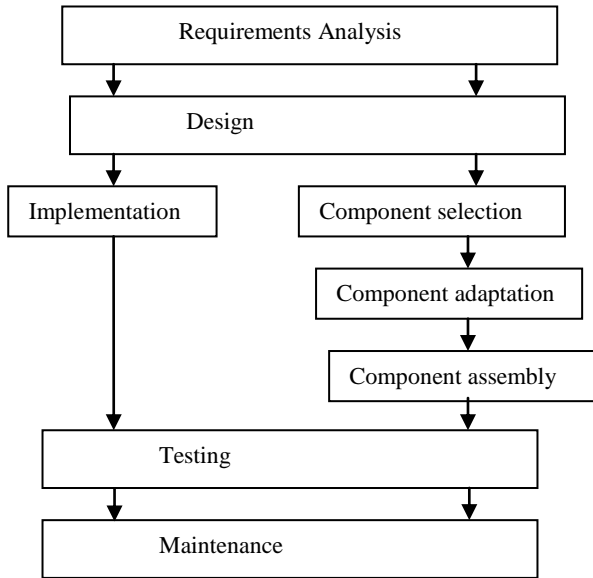
model of traditional software development, and is made up of two Vs, one V for component development and one V for system development. The V for component development involves identifying components from repository or defining components based on domain requirements. The V model for system development involves a process for assembling the components into a system based on system requirements. The W model involves verification and validation for the individual software components as well as for the system.

## 2. Component Based Software Engineering

Software component is a standalone software module having independent functionality which offers its functionality through interfaces. Domain engineering is a part of Component based software development process. In domain engineering, for a domain, some basic functionalities, which are candidates for reuse, are identified and stored in the repository so that they can be used for developing software of that domain. When the software development team gets the requirements for developing software they ask the following questions for each system requirement:

- 1.) Are commercial off-the-shelf (COTS) components available, to implement the requirement?
- 2.) Are internally developed reusable components available to implement the requirement?
- 3.) Are the interfaces for available components compatible within the architecture of the system to be built?

The general process for component based software development comprises of two separate processes: (1) for component development and (2) for system development (using the software components). Component development consists of developing software components for the purpose of storing in the repository so that they can be used for building different software. System development means developing the software using pre-built software components based on the system requirements. Both the processes follow the same life cycle of "Requirements analysis, Design, Implementation, Testing and Maintenance". For component development process, implementation is a single activity whereas for system development, implementation involves component selection, component adaptation and component assembly.



Component life cycle System life cycle

Fig 1 Component based software development process

3. V Model of Software development

V Model of software development is an extension of waterfall model. This model depicts the association between the different phases of software development life cycle and its corresponding phase of testing [1].

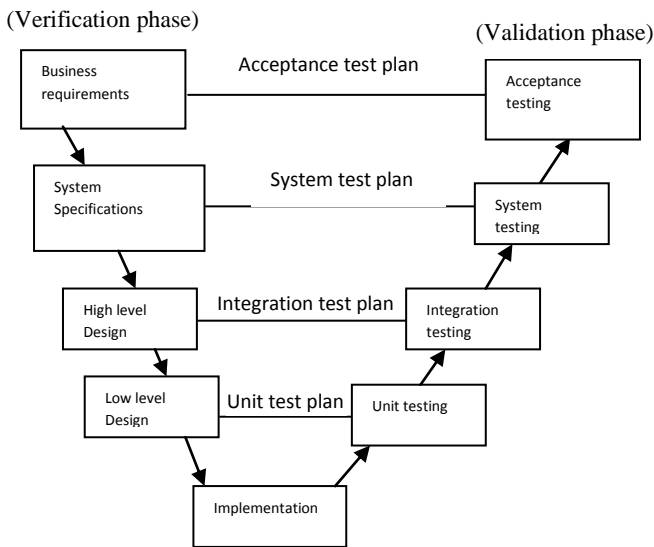


Fig 2 V model of software development

When business requirements are obtained and application developers analyze them, at the same time testing phase is also started and acceptance test plan is made. When developers obtain system specifications from the business requirements then in the testing phase System test plan is made. When high level design is made in the development phase then Integration test plan is made in the testing phase. When low level module design is obtained in the development phase then unit test plan is made. Then actual implementation (coding) takes place. After that, Unit testing is done followed by Integration testing (as modules are integrated) followed by system testing followed by acceptance testing. So, both development and testing phases go hand in hand. In this model, verification and validation take place at each phase of the model.

4. Modification of CBD Model according to V model

The model of Component based software development is essentially a bottom up design approach as the components need to be developed first and then they are selected, adapted and assembled to make the system. In V model top down design approach is followed as the system is designed first and then the modules are designed and implemented.

In V model adaptation of the CBD model the bottom up approach of CBD model is incorporated in the V model and both the component development phase and the system development phase follow the V model. Component design means identifying the functionality of the component along with the interfaces and how it will interact with other components. Component testing means testing the component in the context of the system [1].

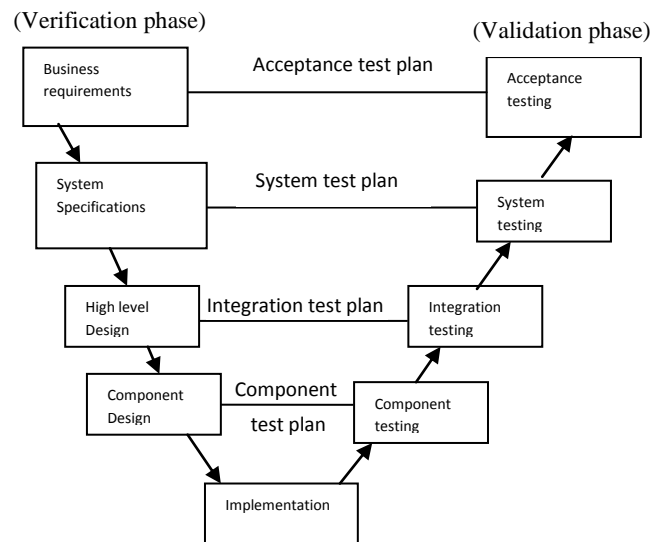


Fig 3 V model adaptation of CBD Model

### 5. W Model of Component based development

The component model considered is X-Man and CBD process based on X-man is as follows [2], [3], [4]:

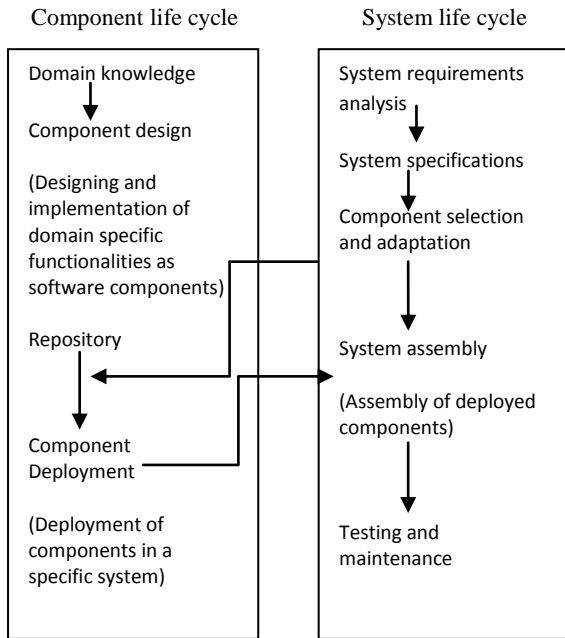


Fig 4. X-Man based CBD model

It consists of a component life cycle and a system life cycle. The component life cycle consists of component design and component deployment, in context of a specific business domain. In the component design phase the specific functionalities of the domain are identified and implemented as standalone software components. These components are then stored in a repository which is domain specific. In component deployment phase the components are retrieved from the repository and deployed into the system being developed.

The system life cycle consists of analysis of system requirements to obtain the set of specifications which are to be implemented in a system. The system design phase consists of component selection (identification of components to be included in the system, from the repository) [1] followed by component adaptation and then component deployment (in component life cycle) followed by system assembly (which is the assembling of all the deployed components to build the final system). The component life cycle links up with the system life cycle with the deployed components being iteratively assembled into the system under construction. The link is depicted by arrows between the life cycles. Components are selected from the repository and then the deployed components are assembled to fit in the system.

Applying Verification and Validation, to component and system life cycles, yields the CBD model with V&V applied to both the cycles. Component V&V, which corresponds to component testing, occurs in component life cycle during the implementation of components. Compositional V&V, which

corresponds to integration testing, occurs in the system life cycle when components are assembled and System V&V also occurs in the system life cycle when the entire system has been assembled.

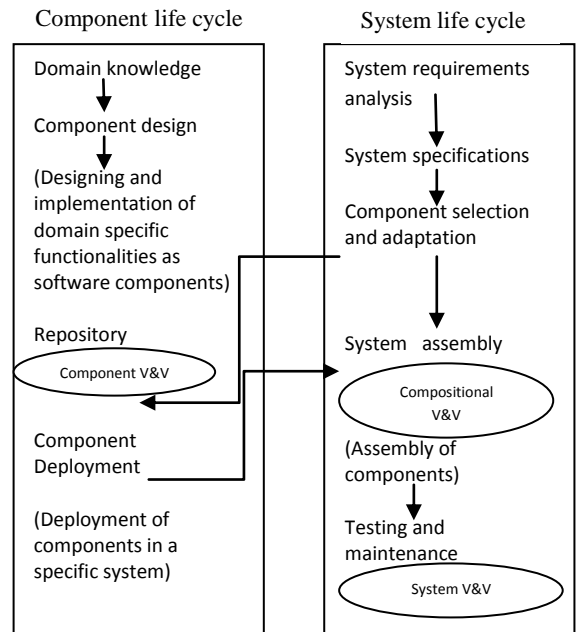


Fig 5. X-Man based CBD model with V&V

X-MAN based CBD model with V&V can be modelled into W model. There are two conjoined V models, one for component life cycle and one for system life cycle. They can be conjoined to form W, hence the name W model.

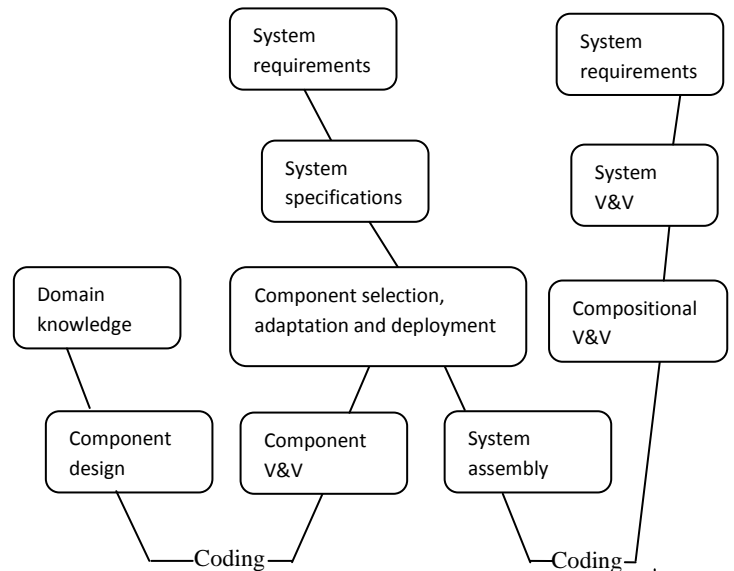


Fig 6. W model of Component based software development

In W model separate life cycles are there for component and system. When for a specific business domain, domain knowledge is gathered and basic functionalities are being identified to implement as components at the same time System requirements are identified and set of specifications

is obtained. After the coding for component is done, component is verified and validated and then selected, adapted and deployed in the context of the system. Then all the deployed components are assembled and coding is done for the interfaces and system is ready. As components are being integrated then compositional V&V is done and once the system is ready system V&V is performed followed by final acceptance testing.

## 6. Conclusion

W model is similar to standard CBD process as both of them contain separate life cycles for component as well as for system. Components are developed using domain engineering techniques and then archived. A framework is then defined for selecting components from the repository and for assembling them into the system. W model contains two V models one for component life cycle and one for system life cycle. This model applies verification and verification at component level as well as system level along with compositional testing which corresponds to integration testing.

## 7. Acknowledgements

The help rendered by friends and colleagues in finalizing this paper is duly acknowledged.

## 8. References

- [1] <http://www.cs.man.ac.uk>
- [2] K. K. Lau, M Ornaghi, Z Wang, A software component model and its formalization, proc, 4<sup>th</sup> Int Symp. on Formal methods for components and objects, LNCS 4111, pages 1-21, Springer-Verlag, 2006
- [3] K.K. Lau and F M Taweel, Domain specific software component models, Proc 12<sup>th</sup> Intl. Symp. On Component based software engineering, LNCS 5582, pages 19-35 Springer-Verlag, 2009.
- [4] P Velasco Elizondo and K.K. Lau "A catalog of component connectors to support development for reuse, The Journal of Systems and Software, 1165-1178, 2010.
- [5] Roger S Pressman, Software Engineering A Practitioner's Approach, 6<sup>th</sup> edition, 2012.