INSIST Vol. 3 No. 2, October 2018 (149–153)
http://insist.unila.ac.id/
DOI:10.23960/ins.v3i2.149

eISSN: 2502-8588

Received: 15/04/2018
Accepted: 16/06/2018
Published online: 20/10/2018

Analysis of lifecycle models and Software Component Retrieval

Monisha Kumari[1],Dimple Nagpal[2]

***Abstract*—**The development of the software component consists of many stages and the lifecycle models show the flow of the stages that the development of the component follows. The development of the new software from the reusable software component has been challenging as it is always a concern to retrieve the relevant software component from the repository. There are lifecycles for making the design and development of the component more efficient. In this paper, an overview of the software component lifecycle models X-model, Y-model, Z-model, knot model and elite model and phases explanation of the lifecycle has been discussed. The current concept explains the comparison of the three lifecycle models in this paper. The component retrieval algorithms has been compared and discussed for the retrieval of the software component.

***Keywords*—**Component Based Development (CBD), Component lifecycle models, Component retrieval, Component reusability, Software component.

I. INTRODUCTION

The component based development deals with the concept of reusing the already existing components. There are number of phases in the development of the software and the construction with the existing components has leads to many changes in the development phases.

The component based development follows various phases in the development.

1.1 Requirements Phase

In[1] C_{BD} the requirement phase consists of the analysis of the specification what the system expected to work. The goals, constraints and component analysis have been the part of requirement phase. This approach can be linked with the example in which obtaining the dining table by order from the carpenter or by purchasing it from the furniture shop. In second case one cannot get the exact result of dining table one wishes to have. In this way the software component gathering that can fit in the software development process is the major part of the requirement phase.

1.2 Analysis & Design Phase

The analysis phase will decide the component that can be fitted in the system development. This phase includes the detailed design of the system in which the relation between the requirements of the system are identifies and described. In C_{BD} the components are designed and developed using framework using the component models which is specified and decided in this phase.

1.3 Implementation Phase

In[1] the implementation phase the design of the system formalize into the executable form. There are two important steps for this phase: 1) Selection of the component 2) Integration of the component with the system. In the first step the component is selected and verified. In second step the integration of components in assemblies tested because invisible dependency that can be resources shared or data shared can make the system fail.

1.4 Integration Phase

In this it checks how the components in the system are integrated and figure out the possibility of integrating of other components in the system for better results.

1.5 Test Phase

In this the software components in the system to be integrated are tested and results of the test are delivered to the software developer along with the component. The components to be tested are stored in the testable component repository.

1.6 Release Phase

The component are delivered and released along with the documentation, test results and test procedures. The components are delivered in the form of package that is suitable for the installation and distribution.

1.7 Maintenance Phase

The maintenance phase includes the repair of the component in the system if the component doing the malfunctioning then it can be replaced with the new component.

1.8 Coding and Archiving

In coding the code consists of steps that have been decided in the design phase written in any programming language which makes the steps understandable to the machine. In archiving the component storage has been involved and figures out the component to be well documented, understandable and well written.

[1]Research scholar,Chandigarh Engineering college,Landran
[2]Faculty,CSE Deptt,Chandigarh Engineering College,Landran

Received: 15/04/2018
Accepted: 16/06/2018
Published online: 20/10/2018

1.9 Select component according software analysis

The selection of the component is done when the specification of the component is close to the requirements of the client. Afterwards selected component undergoes through adaptability and evaluation phase.

1.10 Evaluate & adaptability

The evaluation and adaption is necessary phase in $C_{BD.}$ The components are developed that tends to satisfy the needs of particular software that has been component based. So, to make the component adaptable the developer needs to do some changes in specifications, standards and environments.

1.11 Component wrapping and archiving

The power of the software developer to change the component that can be reused is known as component wrapping. There are three types of component wrapping (see table 1):

TABLE 1
COMPONENT WRAPPING TYPE

Type	Description
W_{BW}	I_P
G_{BW}	C_E
B_{BW}	P_P

W_{BW}=white box wrapping, G_{BW}=grey box wrapping, B_{BW}=black box wrapping, I_P= needs internal processing details of component ,C_E=Application programming interface or component extension language provided by component library,P_P=component interface pre processing and post processing

1.12 Assembly

In this phase the components that are reusable are integrated into the architectural style. The components tend to be managed and interconnected in the appropriate infrastructure.

1.13 Domain engineering

In[2] Domain engineering is the process in which the application domain is analysed to identify the common areas and describing ways with help of uniform vocabulary. It is concerned with the identification of the components that are potential.

1.14 Domain analysis and specification

The components that are extracted from the domain are categorized and investigated in this phase. The classes that can be reused and the classes that are broadly applicable are analysed in this phase.

1.15 Frame working

In frame working the overview of the reusable components and their interrelationships within the application domain has been specified.

II .COMPONENT LIFECYCLE MODELS

The main approach of component based development is building a new application from already existing components. In[1] there are various steps for the development of the system with already existing components (see table 2).

TABLE 2
DESCRIPTION OF COMPONENTS DEVELOPMENT

Steps	**Description**
Step-1	Analysis of already developed component
Step-2	Finding and evaluation of the components
Step-3	Verify changes in component activities

2.1 V Model

V-model is very much disciplined model and beneficial for the small organizations with understandable requirements. The model is based on the waterfall model. V-model follows the bottom up approach for the component development and after completion of one stage the next stage starts. (See Figure 1). The V-model is suitable for the systems with following specifications: 1) Clear, well defined and documented requirements. 2) The project should be small. 3) The requirements should not be undefined or ambiguous. 4) Stable definition of product. 5) Project team easily understands the technology that should not be dynamic.

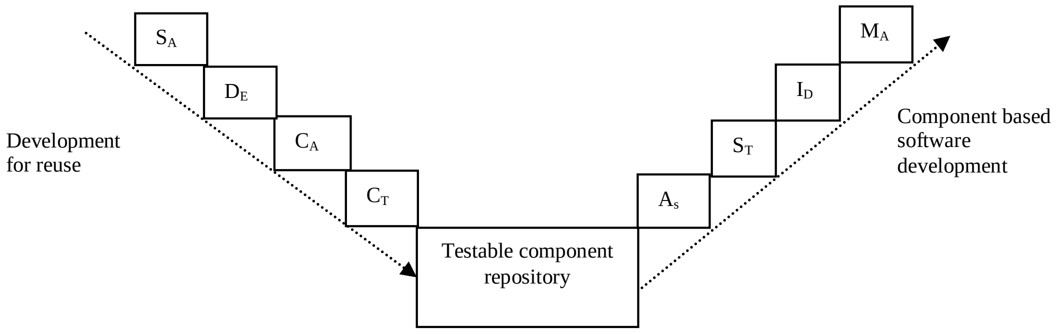
S_A
D_E
C_A
C_T
Development for reuse
Testable component repository
A_s
S_T
I_D
M_A
Component based software development

Fig 1. V-model
S_A=software analysis & specification, D_E= Design, C_A=coding & archiving, C_T=component testing, M_A=Maintenance, I_D= implementation deployment, S_T= system testing, A_S= assembly

Received: 15/04/2018
Accepted: 16/06/2018
Published online: 20/10/2018

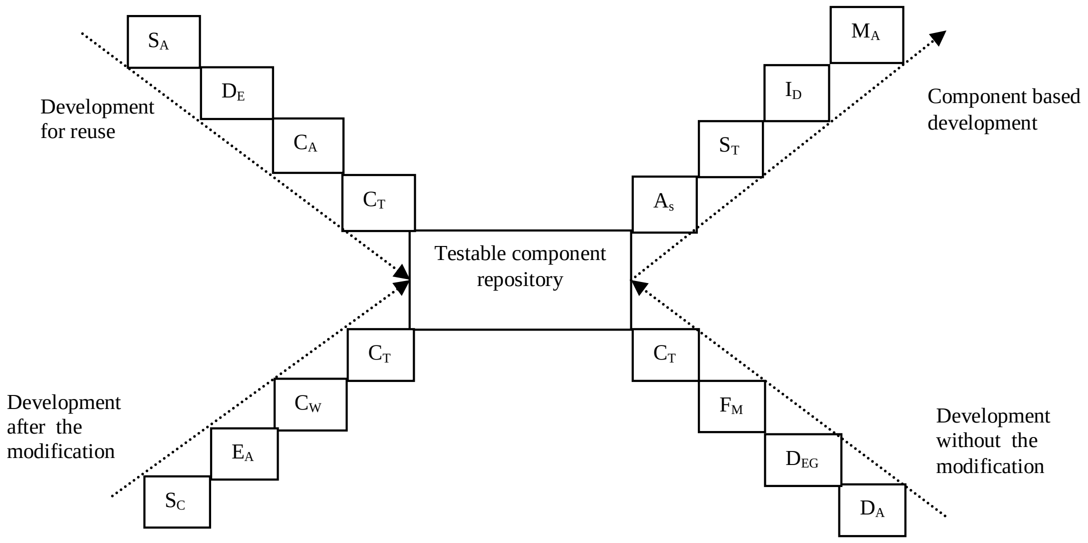
S_A
D_E
C_A
C_T
Development for reuse
M_A
I_D
S_T
A_s
Component based development
Testable component repository
C_T
C_W
E_A
S_C
Development after the modification
C_T
F_M
D_{EG}
D_A
Development without the modification

Fig 2. X-model

S_A=software analysis & specification, D_E= Design, C_A=coding & archiving, C_T=component testing, M_A=Maintenance, I_D= implementation deployment, S_T= system testing, A_S= assembly, C_W=component wrapping and archiving, E_A evaluate & adaptability, S_C=select component according software analysis, specification & design, F_M=frame working, D_{EG}=domain engineering, D_A=domain analysis and specification.

2.2 X Model

In[3] the design of the X models is asymmetrical as the task is divided into constructive work and destructive work. The left hand side would be for constructive work and right hand side would be for destructive work. There are two phases in X-model. First is build of reusable components for development of application. Other one is introduction of build of application from testable as well as reusable component as shown in figure 2.

Testable component repository contains the component after 1) Testing 2) Development for reuse 3) The development with modification and without modification if applicable according to the lifecycle models

In[4] X-model is based on four approaches 1) component development for reuse. 2) Component based software development 3) software development with reusable component modification 4) development of software without any reusable component modification.

2.3 Y Model

In[5] Y lifecycle model mainly focus on the concept of reusability from its core for the development of the software. The components are developed that can be reused for the software development. The reusability is concerned with selecting the component and assembling it to make the software (see figure 3). This model is very much beneficial for the large software development and it takes ideas from both bottom-up and top-down approach. This model can be used for both development for reuse and development with reuse. In this model the previous knowledge of the software developer regarding the application domain is accounted. When the knowledge is limited the bottom-up approach is followed and otherwise most of the times top-down approach is followed. There are two abstraction layers in the Y-model. Component templates are consisted by upper layer and behaviour depicting run time object is consisted by lower layer.

2.4 Knot Model

In[6] this knot lifecycle the reusability is the main focused concept. There are are four phases in the knot model. 1)Pool of reusable component 2) New component development 3) Existing component modification 4) New component based sotware development

INSIST Vol. 3 No. 2, October 2018 (149–153)
http://insist.unila.ac.id/
DOI:10.23960/ins.v3i2.149

Received: 15/04/2018
Accepted: 16/06/2018
Published online: 20/10/2018

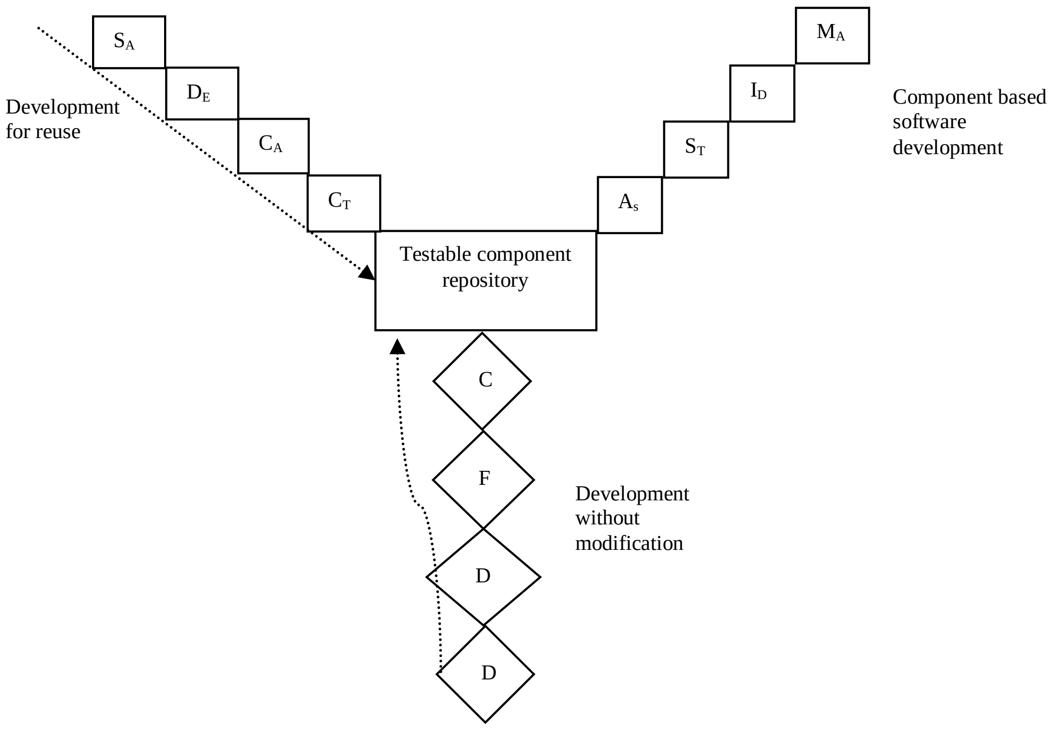
S_A
D_E
C_A
C_T
M_A
I_D
S_T
A_s
Development for reuse
Component based software development
Testable component repository
C
F
D
D
Development without modification

Fig 3. Y-model

S_A=software analysis & specification, D_E= Design, C_A=coding & archiving, C_T=component testing, M_A=Maintenance, I_D= implementation deployment, S_T= system testing, A_S= assembly, F_M=frame working, D_{EG}=domain engineering, D_A=domain analysis and specification.

TABLE 3
PHASES IN KNOT MODEL

Phases	Description
P_R	All the components stored in this including final product and other three phases have direct contact with it.
N_C	Development of component occurs when existing component in the resource pool does not satisfies the requirements of system, design and specification.
E_C	Modification of component occurs when existing component in the pool satisfies the does satisfies the requirement o system design and specification.
N_D	In this phase after risk analysis and testing all the component that are present in the repository are assembled

P_R= Pool of reusable component,N_C= New component development,E_C= Existing component modification,N_D= New component based sotware development

2.5 Elite Lifecycle Model (ELCM)

In[7] the elite lifecycle model is based on the flow of the development of the software based the reusability of the component. There are two fundamental processes that has been shown in Figure 4.

III. COMPARISONS OF LIFECYCLE MODELS

In this paper the comparisons of lifecycle model has been done as in the Table 3 and Table 4.

TABLE 3
COMPARISONS OF LIFECYCLE

Model name	Shape	Pros	Cons
V_M	V_S	E_S, H_D	A_D, H_R
Y_M	Y_S	R_M, B_A	O_P ,R_P
X_M	X_S	A_P,C_E, R_F,	M_C,C_I,L_R
K_M	Ω	R_M,C_E,R_R,L_C	D_S,M_D
E_M	N_A	R_M, C_D, P_E	L_R

V_M=V-model, V_S=V-shape, E_S=Easy and simple to understand, H_D=highly disciplined, H_R=high risk and uncertainty, A_D=Asymmetrical design, X_M=X-model, X_S=X-shape, C_E=Clear explanations of requirements, R_F=Reusability focused, A_P=Avoid overlapping and iterations, M_C=More complexity, C_I=Cost increase, L_R=Lack of risk analysis, Y_M=Y-model, Y_S=Y-shape, R_M=Reusability mainly focused, B_A=both approaches are followed(top down and bottom up), O_P=Overlapping phases,R_P=repetion of process, D_S=Difficulty in Selection of a right component, M_D= huge reservoir and management is difficult, R_R=reduces risk and development time,N_A=not applicable, Ω=knot shape L_C=applicable on larger & complex systems , C_D=Clear design, N_F=not feasible, P_E= perspective on election.

TABLE 4
COMPARISONS OF LIFECYCLE

	V	X	Y	K	E
Development for Reuse	I_n	I_n	I_n	I_n	I_n
Component Based Software Development	I_n	I_n	I_n	I_n	I_n
Development after modification	N_i	I_n	N_i	I_n	N_i
Development without modification	N_i	I_n	I_n	N_i	N_i

I_n=Included, N_i=Not included

Received: 15/04/2018
Accepted: 16/06/2018
Published online: 20/10/2018

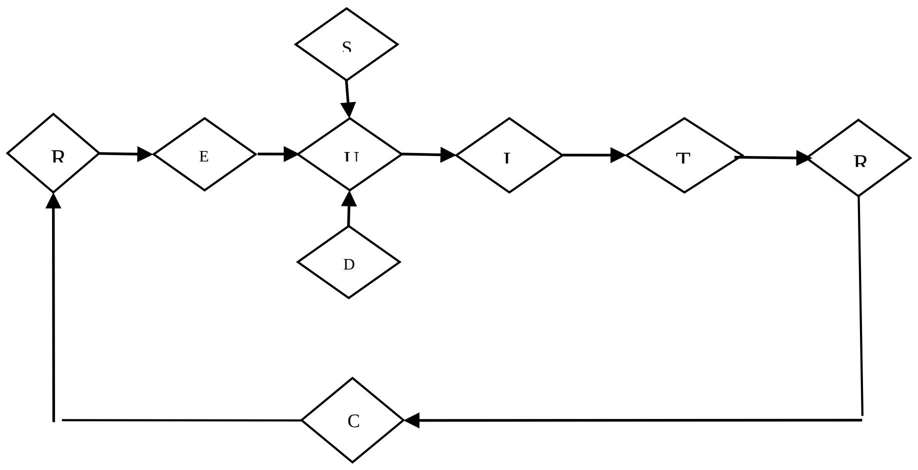

Fig 4. Elite lifecycle model

R_I=requirement indentify,ES= evaluate the scope, UM=use modified and selected component, SC=search the component from existing component repository, D_C=develop new components according to RAD model,T_C=test ,I_C=integrate component as per the elaborated scope specification , R_E=release, C_E=customer evaluation

IV. SOFTWARE COMPONENT RETRIEVAL

The software components have been stored in the repository and retrieval from the repository can be done by using different techniques of retrieval. In[8,9] as the research done in context of component retrieval it has been developed that there are six retrieval algorithms (1) full text retrieval (2) keyword based (3) signature based (4) semantic based (5) Faceted based (6) enumerated based.

TABLE 5
RETRIEVAL TECHNIQUES

S.no	Techniques	Focuses	Dependencies
1	Full text	D_{oc}	Q_{ua}
2	Keyword based	D_{oc}	Q_{ua}
3	Semantic based	S_{tr}	A_v
4	Signature based	S_{tr}	A_v
5	Faceted based	D_{oc}	Q_{ua}
6	Enumerated based	D_{oc}	Q_{ua}

D_{OC}= Documentation of software component, S_{tr}=Structure of software component, Q_{ua}= quality and some pre-processing, A_v=structure availability in form of interface, source code and language processors availability.

V. CONCLUSION

In this paper the evolution of lifecycles model has been discussed. The various phases in the development of the software with the component are explained in this work. The models including X-model, Y-model, Z-model, knot-model and Elite-model have been discussed and comparison of the component model is done in this paper. The component retrieval techniques that have been discussed in this paper can be integrated with artificial intelligence techniques including genetic algorithm, neural networks, and fuzzy and other hybrid techniques for better retrieval of software component.

REFERENCES

[1] Chaudron M, Crnkovic I and Larsson S, Component-based development process and component lifecycle. In ICIT, Information Technology based International Conference ,2005 Aug; DOI: 10.1109/ITI.2005.149119, p.591-596

[2] Tomar P., and Gill NS. Verification & Validation of Components with New X Component-Based Model. In ICSTE, Software Technology and Engineering based International Conference, 2010 Oct, Vol: 2 pp: V2-365 - V2-371, DOI: 10.1109

[3] Tomar , and Gill , New Algorithm for Component Selection to Develop Component-Based Software with X Model. In Software Engineering based Lecture Notes, Aug 2013; 1(3), DOI: 10.7763/LNSE.2013.V1.65.

[4] T. Muhammad, F. Khan, M. Babar, F. Arif, and F. Khan. Framework for Better Reusability in Component Based Software Engineering. In J. Appl. Environ. Biol. Sci.,2016 Mar; 6(4S) pp. 77-81.

[5] Capretz LF, Y: a new component-based software life cycle model. In JCS Journal of Computer Science, 2005 Jan; 1(1) , DOI: 10.3844/jcssp.2005.76.82 p. 76-82.

[6] C R Singh, K Parveen , A New - Knot Model for Component Based Software Development, In IJCS, Int J. Computer Science, 2011 May; 8(3), ISSN : 1694-0814 p.480-484

[7] Bahuguna S., Nautiyal L, Dimri S. C., Tiwari U. K. Elite: A New Component-Based Software Development Model, Int.J.Computer Techology & Applications, 2012 Jan; 3 (1), pp. 119-124

[8] Bakshi, Amandeep, and Seema Bawa. A Survey For Effective Search And Retrieval Of Components From Software Repositories. In IJERT International Journal of Engineering Research and Technology, 2013 April; 2(4).

[9] Hamid /Mcheick., E. Ah-Ki, R. Godin, and Mili, Hafedh An experiment in software component retrieval. In IST Information and Software Technology, 2003 July;45(10) p.633-64