

# Component based Development Methodology for Real time Applications

Vasumathi Devi Majety<sup>1</sup>, GL Sravanthi<sup>2</sup>, K.Chandrakala<sup>3</sup>,Dr.E.Laxmi Lydia<sup>4</sup>  
<sup>1,2,3</sup> Assistant Professor, VNITSW, Guntur, India.

<sup>4</sup>Professor & Big Data Consultant, Computer Science and Engineering, Vignan's Institute of Information Technology, India.

## Article Info

Volume 82

Page Number: 1181 - 1186

Publication Issue:

January-February 2020

## Article History

Article Received: 14 March 2019

Revised: 27 May 2019

Accepted: 16 October 2019

Publication: 06 January 2020

## Abstract

Software development mainly depends on the type of process model which has been selected for specific application. In earlier days object-oriented strategies have been applied but, later on the development process is being done in terms of components development to support reusability. However the predictability of development approach effects the time and cost of the process. This paper projects the different types of process models and also the current trends in the development process. The development methodologies, Behavior Driven Development (BDD) and Test Driven Development (TDD) is emphasized with an example. This work emphasizes components, and its development. The algorithm to develop real time applications is also projected in the work.

**Keywords:** Component , Test Driven Development(TDD) , Behavior Driven Development(BDD).

## 1. INTRODUCTION

The software engineering process models are Water fall model, Incremental process model, Prototyping , Spiral model , Agile process model and Scrum model. Above all these models, a different strategic approach is introduced in the midst of 1970-1975 , that is , Component based software engineering process model. In software development companies now a days , they are following Agile process model and Component based model[2].

Software product is developed by following an approach of Component based software engineering , which emphasizes the design and construction of computer based components which can be reusable in future. This supports the reusability in Object-Oriented programming. These components are used in

some another different application of the same domain[1].

## 2.LITERATURE SURVEY

In software product development time, cost and quality are very crucial metrics. Software development time can be reduced using component based development. The component based development supports the reusability. There by the productivity can be increased. It also reduces the time duration for the completion of the project there by reducing the development cost. But this is not suitable for real time embedded systems development[8].

The comparison between traditional process models and component based development has presented in Table:1. Here development cost , time , man power required , quality and applicability aspects have been observed and presented.

Table 1: Comparison of CBSE with Traditional Process models

Non-funtional parameters	Component Based Development	Traditional Development models
Development Cost	This approach follows reusability in	Cost depends on the application as,

	development process there by reduces development cost	these models does not follow any reusability
Development Time	Time required only to search for the suitable pre-built software (Less time )	The work should start from scratch so the time is comparatively more.
Man power required	Very less in comparison with traditional models	As per the approach of process model , the sufficient man power is required depends on the time duration of project
Quality	The quality is improved as pre-built software it imoprts.	Quality depends
Applicability	This approach of software development is applicable only to software with pre-built software components.	These models are applicable to any kind of application
Note: For most of the real time application development component based development is more suitable than Traditional models.		

### 3. Software Development Life Cycle (SDLC) Activities

A software system is developed by following a set of sequential activities and are called as SDLC activities [2].The system development starts with the communication of customer. This is a very crucial step which initiates the development process. Software Project development steps are defined as Requirement analysis , Design , Coding and Testing as shown in fig.1.,

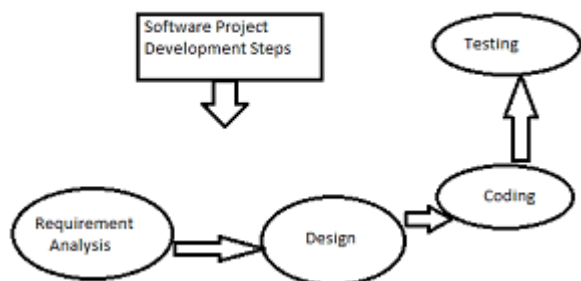


Fig 1: Software Project Development Steps

The project team members should interact with each other and then can come to one final conclusion about the work. The software design follows the Requirement gathering phase.

Designers have to choose the suitable design methodology for the given problem or application. Different types of design strategies are defined such as structure oriented , object oriented , Data structure oriented , modular design , component design etc., The programmers have to strictly follow the coing principles for the smooth functioning of the coding process. After the completion of this entire work the testers will predict the possibilities of error-prone cases and define test cases in testing phase. This entire work is one sequential flow.

### 4. Different Real time applications & Process models

Typical examples of real-time systems include Air Traffic Control Systems, Networked Multimedia Systems, Command Control Systems etc. In a Real-Time System the correctness of the system behavior depends not only on the logical results of the computations, but also on the physical instant at which these results are produced[5]. So, most of the real time applications are following component based development process model rather than traditional process models.

## 5.Component Based Applications

### a) BDD Vs TDD

The component based development methodology is a base for BDD and TDD approaches.

The customer idea is considered as a business deal. Such as an example the customer may place an order for a mobile for his own business. Here the developer may check for the initial requirements such as keypad , touch screen etc., .

He checks for the availability of the components which performs these functionality. If the existing component is perfectly suitable, then only that component is used . Otherwise he has to design a new component with new functionality, which is then be added in Repository for future use. In this instance, the developer will follow component based development for Behavior Driven Development.

### Behavior Driven Development Cycle



Fig: Behaviour Driven Development (BDD)

The Test Driven Development (TDD) , after gathering requirements from the customer , different test cases designed and according to the result of these test cases development process will proceed. Consider the same example ,i.e, if the customer requires a mobile for his own business then by designing test cases like Test case1:

whether user is able to type ; This itself represents the requirement for keypad ; Test case2:Whether the mobile senses the touch ; This itself represents the requirement for touch screen; etc., .Like this the development proceeds by designing different test cases and searching for the right component required.

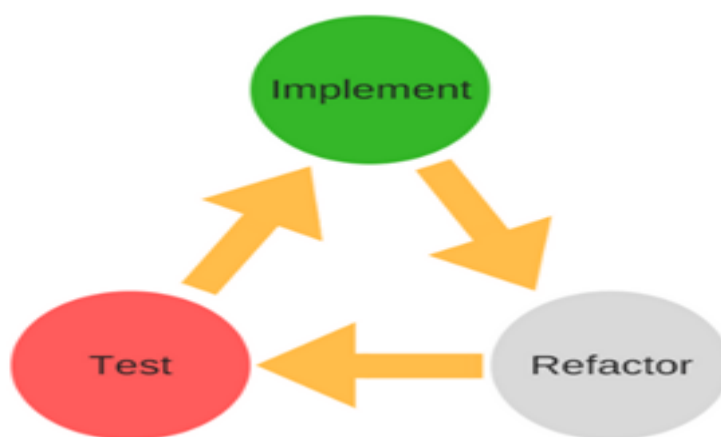


Fig: Test Driven Development(TDD)

### b) Cloud Applications

The purpose of cloud is anyone can design a software and make it available for the public. This itself maintains the reusability. The basic theme behind cloud strategies such as PaaS, IaaS and SaaS is also the same that is, component based development. Here, the component may be a small piece of code, data or Infrastructure. Amazon, Microsoft Azure, Dell EMC cloud service etc., are the examples of different cloud services [15-30].

### c) Component-based software frameworks for specific domains are mentioned below[12]:

- Advanced Component Framework
- Earth System Modeling Framework (ESMF)
- MASH IoT Platform for Asset Management
- KOALA component model developed for software in consumer electronics
- React (JavaScript library)
- Software Communications Architecture (JTRS SCA)

### d) ROBOTICS Application

In Robotics software engineering component based development is used, here a component is a small software or hardware which performs a small part of Robotics functionality. A coordination between software architectures and process models has emerged in the recent days. In robotics components need to be assembled and also decompose the functionality of certain components to redesign a good and large component according to the latest architecture[11]. The main focus is on how to identify the similar kind of components

### e) ANDROID application components

Real time applications are developed using Android. In Android one file named as *Manifest.xml* will maintain the details of all components which are used in the application. Some of the basic components are the same for all applications such as Activities, Services, Broadcast Receivers and content providers. Whereas additional components are Fragments, views, layouts, resources and manifest. In any Android application the major role is by Application component[10].

### 6. Proposed Algorithm

The analyst has to gather requirements from the customer. Among these set of requirements some of the requirements may be already existed. The entire work need not to be start from the scratch. The discovery of right component is the first step for component based development strategy.

#### Algorithm: Component based development algorithm for real time applications

Step1: Begin

Step2: Gather requirements from the customer

Step3: Search for the right component in the component repository

Step4: Initially Sol= null //sol=solution. Initially no components found in solution of the customer application

Step5: If the need of the customer which is represented in terms of customer requirements is similar to the components requirements then consider that component in the 'sol' set.

Step6: If 50% of the user requirements are similar to the components then use first fit strategy and add it to the 'sol' set.

Step7: If 100% of the user requirements are dissimilar to the components specification then start component development from scratch.

Step8: Maintain this newly designed component in component repository for future use.

### 7. Experimenting with the Traditional Approach

The analysis of the different case studies have been experimented in the college for 60 undergraduate students of 6 members per each group. Each team has to come up with innovative idea and has to start the work of development except coding. The prerequisites of coding work is analyzed by each team. We encourage each team to plan and design their case study to make all the students to feel the work of industry. This improves their analytical and logical skills. They analyzed which process model is suitable for the development of their project work.

### 8. Conclusion

Selection of appropriate process model has a high impact in development of real time applications. It improves the quality and cost of development of a software. Most of the nowadays applications follow component based

development rather than traditional process models. In this paper different traditional process model features have been presented in a tabular form in comparison with Component based development. And also component based algorithm has designed to develop a real time component.

## References

1. H. Dogru and M. M. Tanik. A "Process Model for Component-Oriented Software Engineering", IEEE Software, 20(2):34-41, 2003.
2. Software Development Life Cycle (SDLC) – the five common principles.htm
3. Software Methodologies Advantages & disadvantages of various SDLC models.mht
4. Shankar, K., & Ilayaraja, M. (2018, January). Secure Optimal k-NN on Encrypted Cloud Data using Homomorphic Encryption with Query Users. In 2018 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-7). IEEE.
5. [https://users.ece.cmu.edu/~koopman/des\\_s99/real\\_time](https://users.ece.cmu.edu/~koopman/des_s99/real_time)
6. Pressman Roger, "Software Engineering: A Practitioner's Approach", McGraw Hill, Sixth Edition, 2012.
7. Ian Sommerville, "Software Engineering", 8 TH Edition, 2012.
8. IvicaCrnkovic, "Component based Software Engineering for embedded systems", in IEEE proceedings of International Conference on Software Engineering, 2005.
9. <http://www.cs.man.ac.uk>
10. M. L. Alvarez & E. Estévez & I. Sarachaga & A. Burgos & M. Marcos "A novel approach for supporting the development cycle of automation systems", Int J AdvManufTechnol (2013) 68:711–725 .
11. <http://www.techplayon.com/applications-component>
12. DavideBrugali et al., "Component-based Robotic Engineering Part II: Systems and Models" ;IEEE ROBOTICS AND AUTOMATION MAGAZINE, VOL. XX, NO. 1, MARCH 2010.
13. [https://en.wikipedia.org/wiki/Component-based\\_software\\_engineering](https://en.wikipedia.org/wiki/Component-based_software_engineering)
14. M. R.Fox,D. C.Brogan, J.Paul,andF.Reynolds, "Approximating Component Selection,"inProc.the 36thConference on Winter Simulation Conference, 2004, pp. 429-434.
15. Elhoseny, M., Bian, G. B., Lakshmanaprabu, S. K., Shankar, K., Singh, A. K., & Wu, W. (2019). Effective features to classify ovarian cancer data in internet of medical things. Computer Networks, 159, 147-156.
16. Shankar, K., Elhoseny, M., Perumal, E., Ilayaraja, M., & Kumar, K. S. (2019). An Efficient Image Encryption Scheme Based on Signcryption Technique with Adaptive Elephant Herding Optimization. In Cybersecurity and Secure Information Systems (pp. 31-42). Springer, Cham.
17. Elhoseny, M., & Shankar, K. (2020). Energy efficient optimal routing for communication in VANETs via clustering model. In Emerging Technologies for Connected Internet of Vehicles and Intelligent Transportation System Networks (pp. 1-14). Springer, Cham.
18. Elhoseny, M., Shankar, K., & Uthayakumar, J. Intelligent Diagnostic Prediction and Classification System for Chronic Kidney Disease, Nature Scientific Reports, July 2019. Press. DOI: <https://doi.org/10.1038/s41598-019-46074-2>.
19. Dutta, A. K., Elhoseny, M., Dahiya, V., & Shankar, K. (2019). An efficient hierarchical clustering protocol for multihop Internet of vehicles communication. Transactions on Emerging Telecommunications Technologies.
20. Elhoseny, M., & Shankar, K. (2019). Optimal bilateral filter and Convolutional Neural Network based denoising method of medical image measurements. Measurement, 143, 125-135.
21. Murugan, B. S., Elhoseny, M., Shankar, K., & Uthayakumar, J. (2019). Region-based scalable smart system for anomaly detection in pedestrian walkways. Computers & Electrical Engineering, 75, 146-160.
22. Famila, S., Jawahar, A., Sariga, A., & Shankar, K. (2019). Improved artificial bee colony optimization based clustering algorithm for SMART sensor environments. Peer-to-Peer Networking and Applications, 1-9.
23. Lakshmanaprabu, S. K., Shankar, K., Rani, S. S., Abdulhay, E., Arunkumar, N., Ramirez, G., & Uthayakumar, J. (2019). An effect of big data technology with ant colony optimization based routing in vehicular ad hoc networks: Towards smart cities. Journal of cleaner production, 217, 584-593.
24. Maheswari, P. U., Manickam, P., Kumar, K. S., Maseleno, A., & Shankar, K. Bat



- optimization algorithm with fuzzy based PIT sharing (BF-PIT) algorithm for Named Data Networking (NDN). *Journal of Intelligent & Fuzzy Systems*, (Preprint), 1-8.
25. Shankar, K., Ilayaraja, M., & Kumar, K. S. (2018). Technological Solutions for Health Care Protection and Services Through Internet Of Things (IoT). *International Journal of Pure and Applied Mathematics*, 118(7), 277-283.
  26. Lakshmanaprabu, S. K., Shankar, K., Ilayaraja, M., Nasir, A. W., Vijayakumar, V., & Chilamkurti, N. (2019). Random forest for big data classification in the internet of things using optimal features. *International Journal of Machine Learning and Cybernetics*, 1-10.
  27. Sankhwar, S., Gupta, D., Ramya, K. C., Rani, S. S., Shankar, K., & Lakshmanaprabu, S. K. (2016). Improved grey wolf optimization-based feature subset selection with fuzzy neural classifier for financial crisis prediction. *Soft Computing*, 1-10.
  28. Iswanto, I., Lydia, E. L., Shankar, K., Nguyen, P. T., Hashim, W., & Maseleno, A. (2019). Identifying diseases and diagnosis using machine learning. *International Journal of Engineering and Advanced Technology*, 8(6 Special Issue 2), 978-981.
  29. Lakshmanaprabu, S. K., Mohanty, S. N., Krishnamoorthy, S., Uthayakumar, J., & Shankar, K. (2019). Online clinical decision support system using optimal deep neural networks. *Applied Soft Computing*, 81, 105487.
  30. Shankar, K., Lakshmanaprabu, S. K., Khanna, A., Tanwar, S., Rodrigues, J. J., & Roy, N. R. (2019). Alzheimer detection using Group Grey Wolf Optimization based features with convolutional classifier. *Computers & Electrical Engineering*, 77, 230-243.