

# Comparison between Various Software Development Methodologies

Vishal Chandra

## ABSTRACT

From the beginning of software development, it always tried to formulate some process or process sequences or steps to develop or making software. Development of software should be in systematic manner. Today there are variety of software projects some require only good GUI, some require security, some software are made for mission critical tasks, for all these different types of projects require different type of model for development of software. Few well know software development models are waterfall, v-shaped, agile, and object-oriented. All these have their own advantages and disadvantages. This paper try to solve the problem of choosing right methodology for particular software by comparing all software development methodologies. It also figure out the advantages and disadvantages of different methodologies in useful manner not to criticize, so that a particular model will be chosen by an organization or company. This paper divided into two parts first is dedicated to advantage and disadvantages and second part id for comparison between all most popular software development life cycle models. This paper also used for choosing best model for particular project for developing particular software. This paper broadly categorize all software development models into four categories flow based model, iteration based model, object oriented model, structured based model.

## Keywords

Comparison, software development models, advantages, disadvantages, model selection, flow based, iteration based model, object oriented model, structured based model.

## 1. INTRODUCTION

Software development life cycle model is a process by which a software project is completed or developed through the well-defined processes or stages. In present days there are a number of software development models to develop certain software project., so it is very difficult to decide which model is to use for certain type of software project, because every model has its pros and cons, example models take long time to complete, some required less time, some models are for non-technical clients some require well skilled employee, some need crystal clear knowledge about project, one for only developing software from scratch. The success and failure of the project is also depends upon the selection of certain type of software development model. For example for very small and inexpensive project waterfall model is best, some required less human resource, some have time complexity matter, for mission critical projects spiral model is best, for unskilled client prototype model is best, now these days object orient model is popular because it can interact with real world, it does not mean prototype model cannot be used for other projects, it simply means they are good for that type of projects along with other type of projects. Whatever model we choose for developing software projects some mandatory stages are required, without they are as follows: -

1. Feasibility analysis
2. Requirement analysis
3. Design
4. Coding
5. Testing
6. Implementation
7. Maintenance

1. Requirement gathering: - In this phase experts discover whether the proposed software project is feasible or not under technical, economic conditions.
2. Requirement analysis: - In this phase, professionals gather information about proposed project. How much resource, manpower needed for this software project is determined in this stage.
3. Designing:- In this phase we use E-R diagrams, DFDs and other object oriented models to design the software, means how software work. It is very useful for good coding, it is also the blueprint of coding.
4. Coding: - In this phase programming languages ae used to develop or create software. It uses many different languages and software to develop or complete the projects.
5. Testing: - It is the phase in which some testing methods or standards are used to test the developed software using black box testing unit testing integration testing etc.
6. Implementation /Deployment: - after various types of testing, if software pass all testing conditions then it is implemented or deployed in client system.
7. Maintenance:- After implementation organization or company make contract to serve their service of certain time period as maintenance such as adding some component or module or feature, change in platform.

## 2. COMPARISON BETWEEN SOFTWARE DEVELOPMENT MODELS

In this paper all different types' software development life cycle models are divided in four categories.

1. Flow based model
2. Structured based model
3. Iteration based model
4. Object oriented model

Flow model: - There are basically two types of model comes under it:-

1. Waterfall model
2. Iterative waterfall model

### **Waterfall model**

It is one of the classical and primitive model for developing software, in this model there are seven stages to develop the software. It starts with stage one and flow like waterfall means never go back or give feedback. The main disadvantage is that there is no feedback from one stage to other stage, means bugs cannot be detected until and unless whole stages are completed.

#### **Advantages**

- a) It is best for those projects where goals are simple and well understood.
- b) It is very simple.
- c) Stages are well defined.
- d) Easy to implement.
- e) End users are also well known about projects.
- f) Time consuming.

#### **Disadvantages**

- a) Not flexible
- b) No feedback facility to rectify errors
- c) Until and unless one stage is complete one cannot go to next stage
- d) Not good for complex and mission critical projects.

### **Iterative waterfall model**

It is the modified form classical waterfall model. It has facility of giving feedback to the previous stage, with the help of this facility it is easy to rectify error before going to next stage. Since error propagate to backward that's why it is called iterative waterfall model.

#### **Advantages**

- a) Less flexible.
- b) It is very simple.
- c) Stages are well defined.
- d) Error rectification from one stage to other can be possible.
- e) Best for well understood goals.
- f) Easy to implement.

#### **Disadvantages**

- a) Not good for complex and mission critical projects.
- b) Within completion of one stage second stage will open.
- c) No iteration after implementation.
- d) Time consuming.

### **Structured based model**

In this categories there are two models

- a) Spiral model
- b) V model

### **Spiral model**

It is said to a structured model because it has well defined structure for developing software. It is also known as Meta model. It is best suited for complex and mission critical projects. Each trip around the spiral traverses four basic quadrants:

- (1) Plan the next iteration.
- (2) Determine objectives, alternatives, and constraints of the iteration.
- (3) Develop and verify deliverables from the iteration. And
- (4) Evaluate alternatives; Identify and resolve risks.

#### **Advantages**

- a) Software development is divided into smaller parts and risky parts.
- b) Requirement change during development can be accepted.
- c) Best for critical software development.
- d) It uses more prototypes.

#### **Disadvantages**

- a) It is very complicated.
- b) Goals must be well understood.
- c) Developers as well qualified and experienced with this type of projects.

### **V model**

It is also a structured model, because it has v shaped structure. In this model system requirement and system testing run along with each other, software requirement runs with software testing, software design along with integration testing, and software implementation with unit testing.

#### **Advantages**

- a) It is very versatile model and flexible.
- b) Consume less time.
- c) Required more skilled and experienced employee
- d) Requirements may change during development phase.

#### **Disadvantages**

- a) Not flexible rigid like waterfall
- b) Not suitable for small projects.
- c) Stages and phased are not so clear and well understood.

### **Iteration based model**

There are basically three types of models comes under it they are as follows:-

- a) Prototype model

- b) Evolutionary model
- c) RAD model

### Prototype model

This model is said to a prototype model, because at first prototype model is made after applying testing stage it is give it to the client, if client satisfied then again give it to developer to add functionalities.

#### Advantages

- a) It is best when goal is not well understood
- b) It is useful when client is not technically not well.
- c) Requirement may be added while developing software.
- d) It is most flexible model for developing software.

#### Disadvantages

- a) It is more time consuming model
- b) Required more money
- c) Not good for big projects

### Evolutionary model

This is also said to iterative model because features are incremented one by one to create full software. It is like evolution of something. First create it structure then add component one by one, at any stage system is alive means system works but limited functionality.

#### Advantages

- a) It is well suited when goal is not well understood.
- b) It more flexible model
- c) It is easy to implement
- d) Progress can easily measured

#### Disadvantage

- a) It is time consuming
- b) Require more money
- c) Difficult to get continuous evaluation

### RAD model

It is said to be a RAD model because as requirement increases it again deployed to the system or software. It almost the combination of incremental model and evolutionary model.

#### Advantages

- a) Require less time
- b) Change in requirement can be added
- c) Require few manpower
- d) Time to time progress can be evaluation is possible

#### Disadvantages

- a) Not suitable for large projects
- b) Possibility of high risk
- c) Less scalable

### Object oriented model

Now these days this model is very popular, this model maps the problem with real work, object is any real world entity.in this model everything is represented as class and object. Problems are solved with the help of class diagram, object diagram, activity diagram etc.

#### Advantages

- a) Problem solving approach is good
- b) Real world problem mapping
- c) Easy to understand
- d) Suitable for both small and large project

#### Disadvantages

- a) Difficult to implement
- b) Required more resources
- c) Required skilled developers

Table 1. Comparison Table

Model\ feature	Classical Waterfall	Iterative waterfall	Spiral Model	V model	Prototype model	Evolutionary model	RAD model	Object oriented
flexibility	No	No	Few	Highly	Fully	Fully	highly	More
Cost	Low	Low	High	Medium	High	High	More	More
Risk factor	High	High	No	Less	Less	Less	More	More

<b>Simplicity</b>	Very	Very	Complex	Little	Little	Little	More	Not
<b>Development Time</b>	Little	Little	More	High	High	High	Few	More
<b>Expertise required</b>	No	Not	Highly	Little	Little	Little	More	Highly
<b>Phase containment of error</b>	Yes	Yes	No	Little	No	No	Little	No
<b>Success and failure ratio</b>	Low	Low	High	Medium	High	High	High	Medium
<b>Client interaction</b>	one time	One time	Some	Frequent	Frequent	Frequent	Some	One time
<b>Time required</b>	Few	Few	More	More	More	More	Little	More
<b>Resource required</b>	Few	Few	More	More	More	More	Few	more

### 3. CONCLUSION

Here in this paper there are seven software life cycle models are compared. This paper is not only for student or researchers but also for team leader or team manager to decide which model to choose for certain type of project at Software Company to develop software. Classical waterfall and iterative waterfall model are not so good for any type of project. Spiral is best for high risk projects or mission critical projects. Evolutionary best for not well understood projects or goals. Object oriented model is new and popular, but it is difficult to map with real world.

### 4. REFERENCES

- [1] Kling, R., and W. Scacchi, the Web of Computing: Computer Technology as Social Organization, Advances in Computers, 21, 1-90, Academic Press, New York, 1982.
- [2] Winograd, T. and F. Flores, Understanding Computers and Cognition: A New Foundation for Design, Ablex Publishers, Lexington, MA, 1986.
- [3] W.W. Royce, "Managing the Development of Large Software Systems: Concepts and Techniques", IEEE, IEEE Computer Society, August 1970, pp. 1-9.
- [4] B.W. Boehm, "A Spiral Model for Software Development and Enhancement", IEEE, IEEE Computer Society, vol. 21, issue 5, May 1988, pp. 61 – 72.
- [5] Ian Sommerville, "Software Engineering", 8th Edition, 2006, pp.89.
- [6] R.J. Madachy, "Software Process Dynamics", New Jersey: Willey Inter science, 2007, pp. 31.
- [7] R.S. Pressman, "Software Engineering, A Practitioner's Approach", 5th Ed. New York: McGraw-Hill, 2001, pp. 32.
- [8] E.I. May, B. A. Zimmer, "The Evolutionary Development Model for Software", Hewlett-Packard Journal, Article 4, August 1996, pp.1-8.
- [9] B.W. Boehm, "Anchoring the Software Process", IEEE, IEEE Software, vol. 13, issue 4, July 1996, pp. 73-83.
- [10] R.J. Madachy, "Software Process Dynamics", New Jersey: Willey Inter science, 2007, pp. 33.
- [11] R.S. Pressman, "Software Engineering, A Practitioner's Approach", 5th Ed. New York: McGraw-Hill, 2001, pp. 34.
- [12] P. Kruchten, "Rational Unified Process Best Practices for Software Development Teams", Canada: rational Software, 2001.
- [13] Jeff Tian, Southern Methodist University, Dallas, "Software Quality Engineering", IEEE, Computer Society Publication, Willy Inter Science, 2005.
- [14] Rajib Mall, "Fundamental of software engineering "2nd edition. Prentice-hall, pp. 101-102.