Volume 4, Issue 6, June 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Comparative Analysis of Software Development Process Models

1 Dr. B V Ramana Murthy,
Department of CSE,
Jyotishmathi College of Technology, and Science, Shamirpet, India

2 Prof. Vuppu Padmakar
Department of CSE
Chilkur Balaji Institute of Technology, Aziz nagar, Hyderabad, India

3 Ms. A. Vasavi,
Dept. of CSE
JCTS, Shamirpet
Hyderabad, India

Abstract: Software engineering is the establishment and application of sound engineering principals for obtaining economically feasible and reliable software that can run efficiently on any real time machine. The main objective of software engineering is to develop and deliver high quality software which basically is an intangible element that has no weight, volume or color. Software processes is an integral part of software engineering which helps in the development of product/software. Different software processes are available and each has it own significance depending on the type of the project. Software process models are the step by step activities to be taken out for the development of product. The main purpose of software processes model is to attain the feature within cost, within time and with good quality. Different process models are available in the industry; right process model is administered depending on the type of project, domain of type, size of project and so on. An attempt has been made to take out the comparative study of software processes model by signifying out the number scale from 0-5, influence increase from 0 to 5 on activity and process models.

Keywords
Software Engineering, Software cost, Universal models. Agile model, feature of software projects.

I. Introduction

The Software Process Model defines a distinct set of activities, actions, task, milestones and work product that required to develop quality software. Process model describes a road map for software engineering work. The process model provides stability, control and organization to an activity. The process guides a software team through a set of framework activities that are organized into a process flow that may be linear, incremental or evolutionary.[2]
The software process models are listed:

II. Waterfall Model or Classical Life Cycle Model:

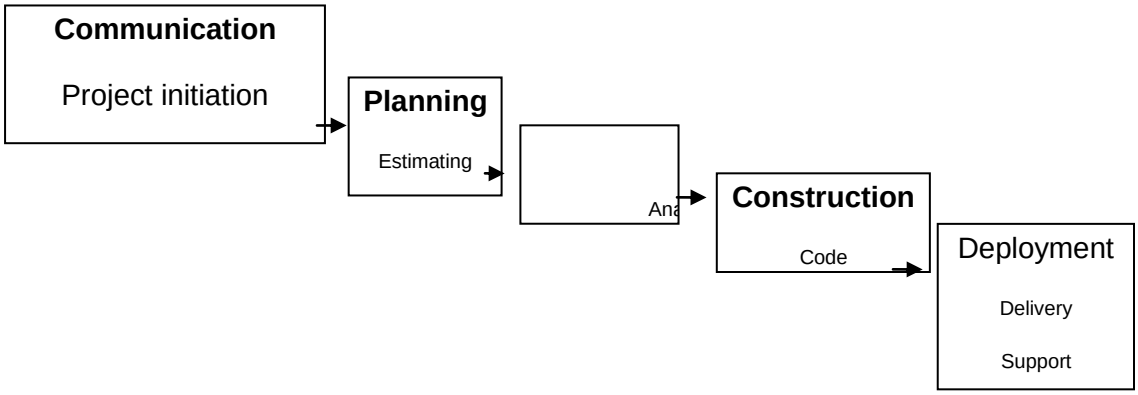
It suggests a systematic, sequential approach to the software development that begins with customer specification of the requirements and progress through planning, modeling, construction and deployment. [1]
Phases of Water Model:
a) Communication:
It is concern with the project initialization and requirements gathering for the development of the project.
b) Planning:
It emphasis on the estimation of Cost, Schedule and other related thing for the development of the project
c) Modeling:
It is concern with the analysis and design of software development.
d) Construction:
It converts the design into coding with help of programming Language and after the completion of the coding then testing will be performed.
e) Deployment:
It is concern with delivery of the software to the customer and customer will provide feedback to the developer.

Advantages of Waterfall Model:

1) It provides for baseline Management, which identifies a fixed set of documents produced as a result of each phase in the life cycle.
2) It is a sequential process for the development of software [3]

Limitations of Waterfall Model:

1) Real projects rarely follow the sequential flow. In these changes results in confusion as the project team proceeds [6]
2) It is often difficult for the customer to the state all requirements at ones
3) A working version of the program will not be available until late in the software development.

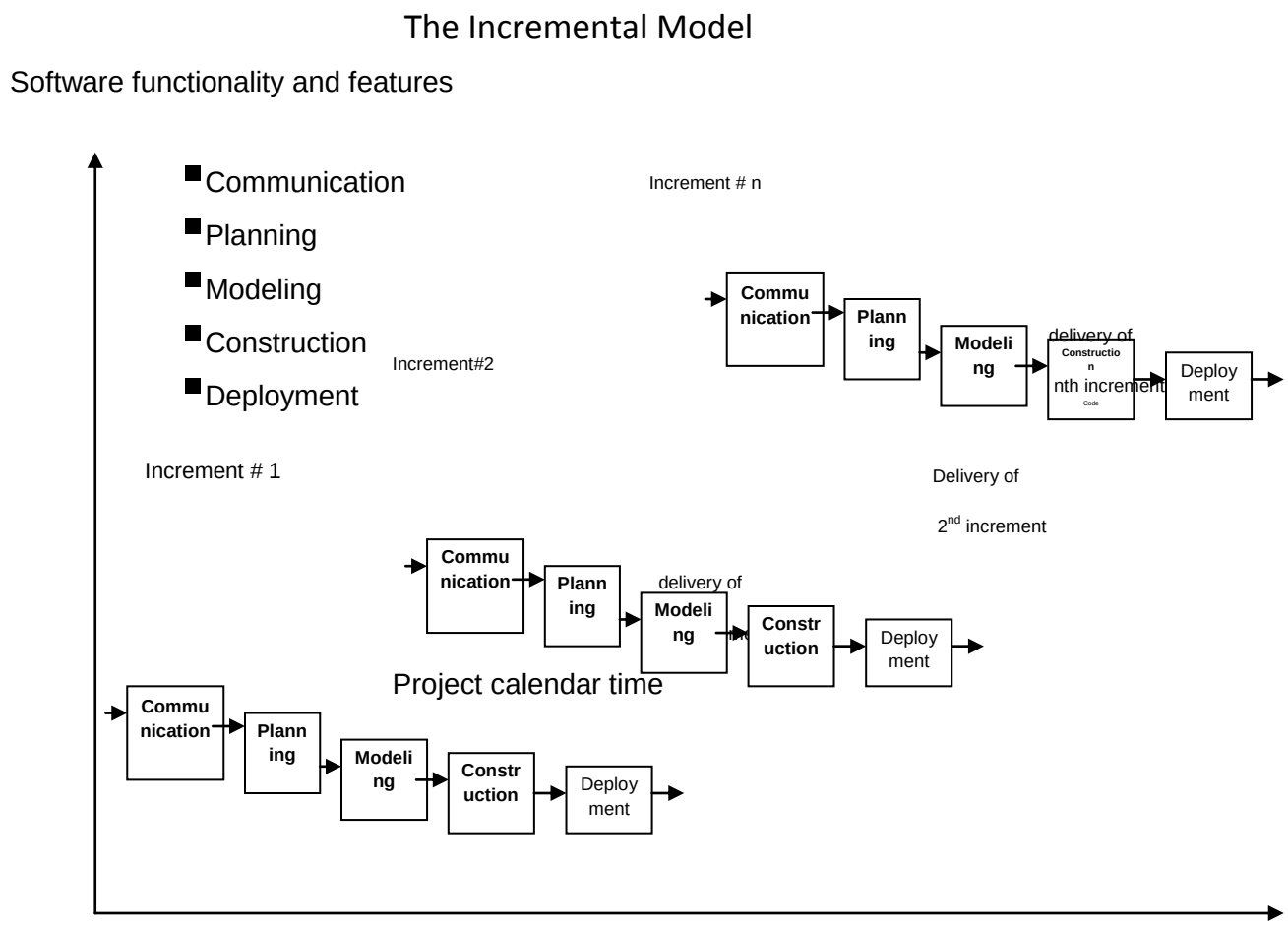


III. Incremental Process Model:

In these model, the system and the software concepts and requirements are first identifies and then remaining activities of the software development are repeated each time there is a new release of the software.

It consist of two models
1) Incremental Model
2) Rapid Application Development model.
Incremental Model :
The incremental model combines the elements of the waterfall model applied in an iterative fashion .Each linear sequence produces a deliverable increment of the software.

In this model the first increment is often called a core product. The core product is used by the customer as a result of the use or evaluation a plan will be developed for the next increment. This plan addresses the modification to core product to meet the needs of the customer these process is repeated until the complete product is produced



IV. Rapid Application Development model (RAD)

RAD is an incremental software process model that emphasizes a short development cycle. It is a high speed adaptation of the waterfall model. If the requirements are understood and project scope is constrained then RAD team creates a fully functional system in with in a short time period example (60 -90) days.

Advantages of RAD Model:

This model is used to develop software in the fastest manner.

2 These model is used for large software development

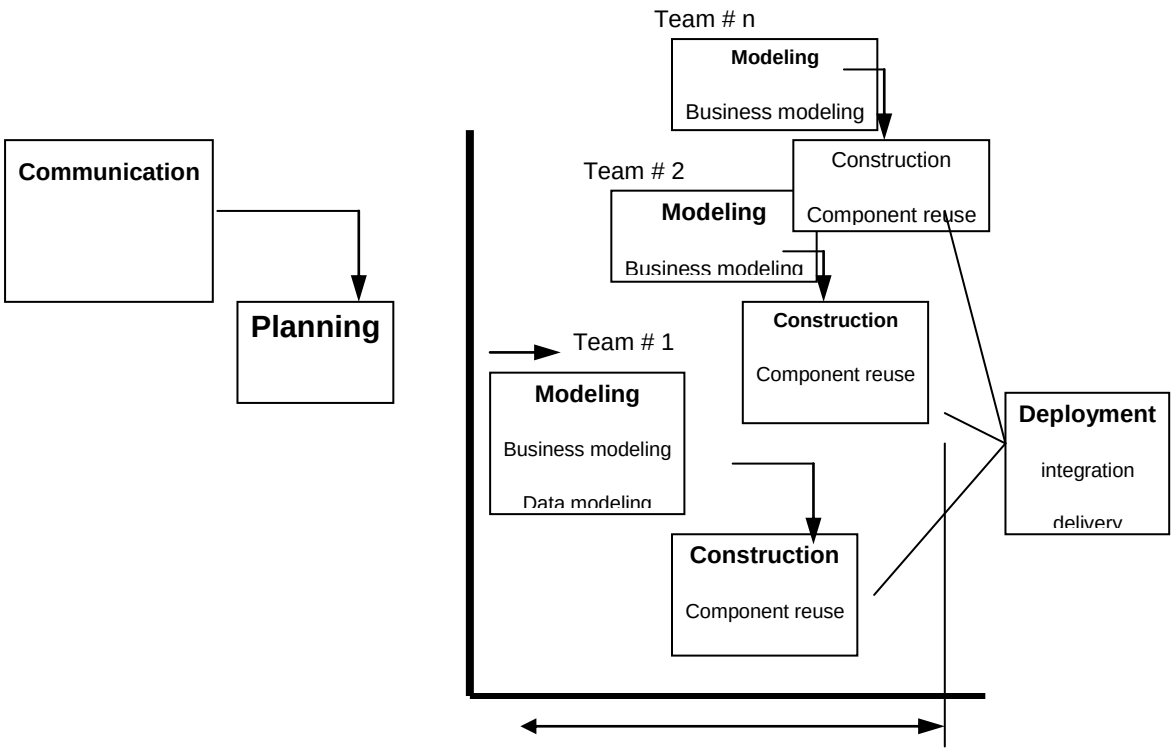
Limitations of RAD model:

1 It is used for large projects and not for small projects.

2 If developer and customer is not committed for the rapid fire activities then RAD model is not used

3 RAD may not be appropriate when technical risk is high

The RAD Model



Evolutionary process model:

Evolutionary process model are iterative. They are characterized in a manner that enables software engineers to develop increasingly more complete version of the software.[4]

It consists of three models:

1) Prototype Model
2) Spiral Model and
3) **win –win spiral model**

V. Prototype Model:

When a customer is not identify detail input processing and output requirement and if the developer may be ensure of the efficiency of an algorithm or adaptability of the operating system. In the above situations prototype process model is used to assist the software engineers and the customers to better understand what is being developed when requirements are fuzzy.

The prototype model begins with communication [1]. The software engineer and the customer meet and define the overall objective of the software then prototype iteration is planned quickly and then modeling occurs.

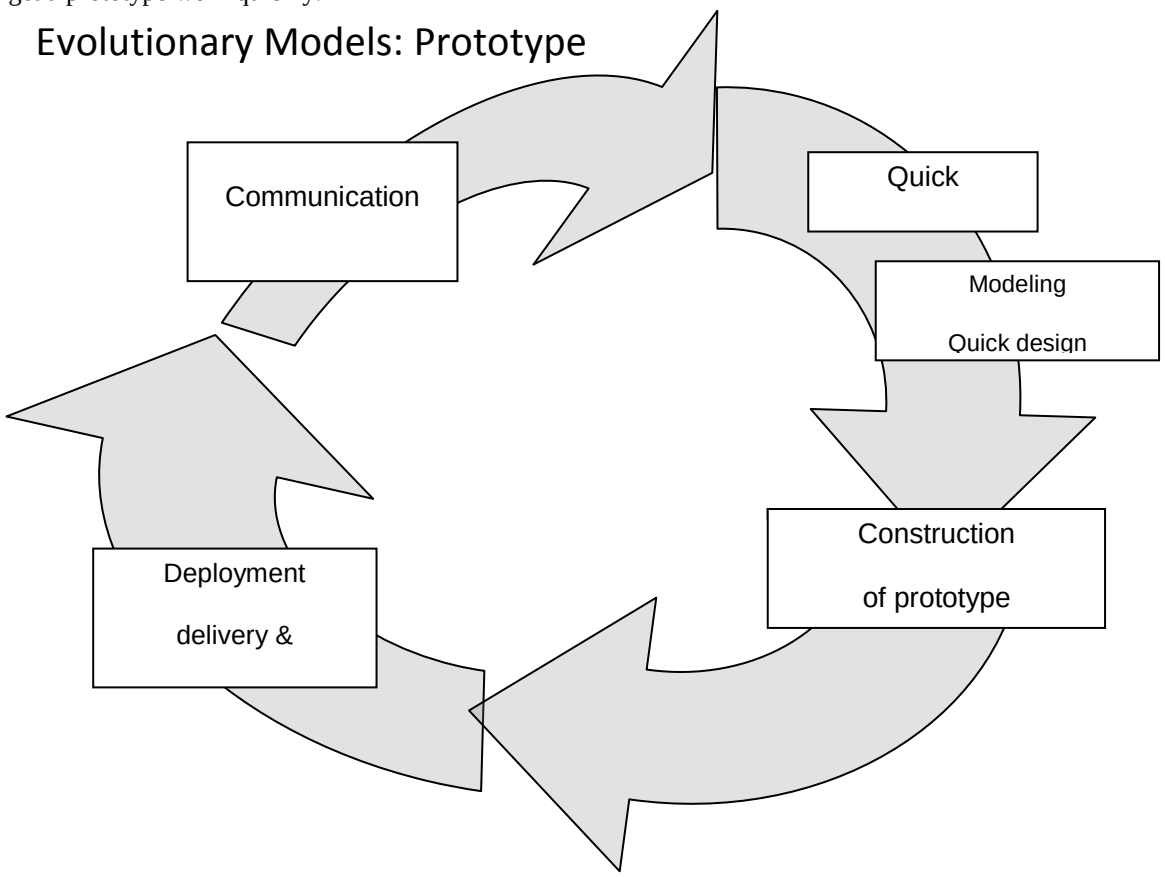
The quick design leads to construction of the prototype then the prototype will be evaluated by the customer and feedback of the customer is used to refine the requirements iterations occurs as the prototype is tuned to satisfy the customer while at the same time enabling the developer to better understand the what needs to be done.

Advantages of Prototype model:

1. It first develops the working version of the programs.
2. The customer understand the software development process easily
3. Customer gives the feedback for the next increment of the prototype

Limitation of Prototype Model:

1. The customer sees what appears to be a working version of the software, unaware that the prototype is incomplete.
2. The developer is often makes implementation compromises in order to get a prototype work quickly.



VI. Spiral Model:

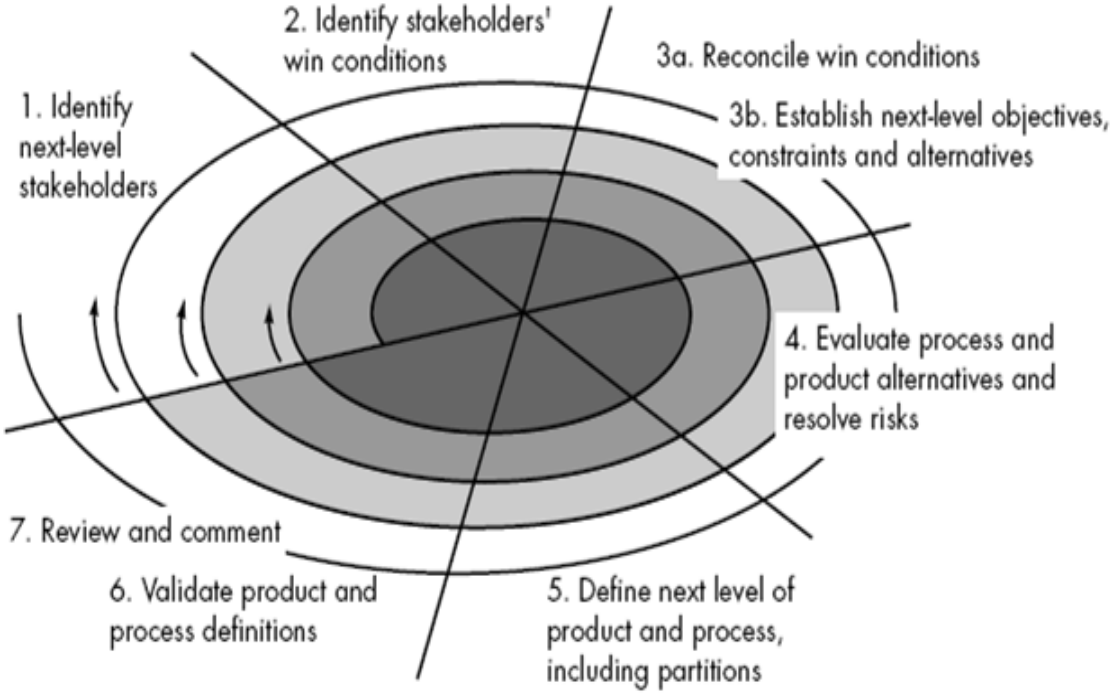
The spiral model is the iterative nature of the prototyping with the controlled and systematic aspects of the water model. In the spiral model software is developed in a series of the evolutionary releases.[6] A spiral model is divided into a set of frame work activities defined by the software engineering team. The software team performs the activities tat are implied by a circuit around the spiral in the clockwise direction, beginning at the center.

The first circuit around the spiral might result in in the development of the product specification. Each pass through the planning region results in the adjustment to the project plan, cost and schedule are adjusted based on the feedback from the customer

The first circuit around the spiral might represent the "concept of the development project" which starts at the center of the spiral and continues for multiple iterations until the concept of development is complete. In spiral model risk is considered at each activity of the software development.

The spiral model is used for large scale systems and software.
Advantages of spiral model:
1. It is an iterative and interactive development of the software.
2. It consider risk analysis at each phase of the spiral model
3. It terminates a project if it is too risky.

Limitations of Spiral Model:
1. Origin of objective, constraints and alternatives are unclear in the spiral model
2. It does not define the organizations major milestones for development of the software



Win – win Spiral **model**:
The win – win spiral model has a provision for the system stakeholders (example customers, developers maintainers testers and general public) to hold a unique mutually satisfactory specification. [3] An industrial grade win – win groupware tool is being developed by terming it as win –win negotiation model. This model facilitates negotiation of the mutually satisfactory system specification by distributed stakeholders.
An agreement with the stakeholders is accompanied by a fundamental reason. These agreement covers win–win condition. During negotiation stakeholders adopt an option relative to an agreement. Associated with each option in an issue relative to the win conditions
Advantages of win – win Spiral Model:
1) It involves stakeholders in the development of the software
2) It provides the basis for a unified effort in tackling the essential difficulties of the software development.

Limitations of Win – win spiral model:
1) The win – win spiral model doesn't address the issues of how developers specify and design.
2) It doesn't specify the conceptual structure of the software.

Comparative Analysis
A comparative study of software process model has taken out with numerical scaling which depicts the influence on feature vs process model. The below comparative study is taken out on discussion and research to find out optional work schedules to be adopted for the software development life cycle depending on the nature of the project. Scaling influences from 0 to 5

- 0 – No influence
- 1 – trivial influence
- 2 – less influence
- 3- modern influence
- 4 – strong influence
- 5 – very strong influence

Feature/ Model	Waterfall	Prototype	Incremental	Spiral	RAD
Requirement Specifications	3	2	3	3	2
Understanding Requirements	5	3	4	3	3
Development Cost	4	3	3	3	2
Guarantee of Success	2	4	4	4	3
Risk Involvement	4	2	2	2	1
Expertise Required	5	3	3	3	2
Changes Incorporated	5	3	4	2	4
Overlapping phases	5	2	4	3	4
Flexibility	5	4	4	3	3
Maintenance	5	2	2	2	2
Reusability	5	2	2	3	4

VII. Conclusion

The software process model differ in the amount of detail with the software process is described, the number of work product products that are identified and used in the process, the interdependency factor, project tracking and control, requirement engineering, glenity in work structure with represent to product development and the above table represents that the development of product vs process model which depicts that for the project which has complete requirements and analysis waterfall model is advisable because it does not have back tracking and it is linear in nature any change to the product it is difficult to incorporate. Similarly a product with incomplete requirements and less in size RAD model is preferable than the other models. The comparative study will help to choose right process model to the project specification. Software processes model is core for software development life cycle in a phased phase. New processes model can be evolved by research depending on the type of projects, domain of the project and client specification. Software processes model can be tailor made to software projects and requirements. Many organizations has their own processes model to taken.

References

[1] Yourdon, e. “When good enough is Best”, IEEE software, vol.12 no.3, may 1992, PP 79-81
[2] Gibb, T, Principals of software engineering management, addrison – wesly 1988
[3] Hanna, M, “Farewell to waterfalls” software magazine may 1995 PP 38-46
[4] Butle J, “Rapid Application Development in action”, Managing system development, applied computer research, vol.14 no.5, may 1994, PP 6-8.
[5] Bochm, B, “A spiral model for software development and enhancements” computer, vol. 21, No5 , May 1998, PP 61-72
[6] Bochm B, “Using win-win spiral model : A case study,” Computer Vol. 31, No7, July 1998, PP 33-44.
[7] Rothi, J.,Yen, D, "System Analysis and Design in End User Developed Applications", Journal of Information Systems Education, 1989.
[8] Fowler, M. (2000), "Put Your Process on a Diet", Software Development”

