Comparative Study of Various Process Model in Software Development

Ratnmala R. Raval
Computer Department
School of Engineering
RK University,
Rajkot, Gujarat

Haresh M. Rathod
Computer Department
School of Engineering
RK University,
Rajkot, Gujarat

ABSTRACT
The software engineering process can be considered at two distinct levels: At the start or first level activities related to the gaining information, development, and maintenance of software; in next stage or second level the activities related to the dentition, functioning, measurement, and upgrading the software process itself. this paper present comparative study of various process models in software development based on various parameters; also listed various factors for choosing partial software model in the world of software development.

Keywords
Introduction of Process Model, Comparative Study of Process Model with Different parameter, factors affecting to Chose Process Model.

1. INTRODUCTION
Software systems starts to and go through a chain of route that report for their beginning, initial development, dynamic operation, keep, and leaving from one production to another. This research sort out and examine a various methods for describing or model how software systems are developed. At the very first level its starts with background and categorization of famous software life cycle models which are given in various software engineering text books.

2. SOFTWARE PROCESS MODEL
A software process model is an abstract representation of processes which are using to develop the software. it simply follow the sdlc methodology which includes Analysis, Design, Implementation, Testing, and Maintains.

2.1 Waterfall Model
The waterfall model is the classical model of software engineering. This model is one of the oldest models and is widely used in government projects and in many major companies. As this model emphasizes planning in early stages, it ensures design flaws before they develop. In addition, its intensive document and planning make it work well for projects in which quality control is a major The pure waterfall lifecycle consists of several non-overlapping stages, as shown in the following figure. The model begins with establishing system requirements and software requirements and continues with architectural design, detailed design, coding, testing, and maintenance. The waterfall model serves as a baseline for many other lifecycle models.

2.2 V-Shaped Model
It work same as waterfall model, the V-Shaped life cycle is a chronological lane of implementation of processes. Each phase is compulsory to complete before the next phase begins. Testing is highlighted in this model more than the waterfall model. The testing actions are developed early in the life cycle before any coding is done, during each of the phase's previous implementation. Requirements start the life cycle model as the waterfall model. Earlier than development is started, a system test plan is created. In this model testing is spotlighted on meeting the functionality specified in requirements gathering. The main design phase spotlight on system architecture and design. The combine test plan is created in this phase in order to test the pieces of the software systems capacity to work jointly. Though, the low-level design phase lies where the real software components are designed, and unit tests are created in this phase. The accomplishment phase is, again, where all coding is generated. After coding is complete, the way of execution continues up the right side of the V where the test plans developed earlier can use.

2.3 Incremental Model
It Combines elements of the waterfall model applied in an iterative fashion each linear sequence produces deliverable "increments" of the software. The first increment is often a core product. The core product is used by the customer (or undergoes detailed evaluation) Based on evaluation results, a plan is developed for the next increment. The incremental process model, like prototyping and other evolutionary approaches, is iterative in nature but unlike prototyping, the incremental model focuses on the delivery of an operational product with each increment particularly useful when Staffing is unavailable. Increments can be planned to manage technical risks

2.4 Spiral Model
The spiral model is same as to the incremental model, with more highlighting on risk analysis. The model divided in four phases: Planning, Risk Analysis, Engineering and Evaluation. A software project frequently passes through these phases in iterations (called Spirals in this model). At the initial spiral, starting with the planning, requirements are gathered and risk is considered. Each consequent spiral builds on the initial spiral. Requirements are collect during the planning phase. In the risk analysis phase, a process is going on to identify risk and their alternate solutions. A prototype is produced at the end of the risk analysis. The evaluation phase permits the customer to assess the output of the project to date before the project goes to the next spiral.

2.5 Rad Model
Its Rapid Application Development model. One type of incremental model. In this model the components or functions are developed in parallel as if they were mini projects. The development are time boxed, delivered and then bring together into a working prototype. This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements.

2.6 Extreme Programming

An approach to development, based on the development and delivery of very small increments of functionality. It relies on constant code improvement, user involvement in the development team and pair wise programming. It can be difficult to keep the interest of customers who are involved in the process. Team members may be unsuited to the intense involvement that characterizes agile methods. Prioritizing changes can be difficult where there are multiple stakeholders. Maintaining simplicity requires extra work. Contracts may be a problem as with other approaches to iterative development.

2.7 Agile Model

Agile development is one kind of iterative and Incremental development model. Where Software is developed in incrementally, rapid cycles. Which gives in small incremental releases with each and every release building on previous functionality? In every each release is thoroughly tested to ensure software quality is maintained or not. Which used for time critical applications. Extreme Programming (XP) is now a day's one of the most well known agile development life cycle model .another methods are scrum, crystal Methodology ,Mobile-D[9]

2.8 XP and Agile principles

1. Incremental development is supported through small, frequent system releases.

2. Customer involvement means full-time customer engagement with the team.

3. People not process through pair programming, collective ownership and a process that avoids long working hours.

4. Change supported through regular system releases.

5. Maintaining simplicity through constant refactoring code

There are some other models are exist which is not much important to be consider prototype model which include user feedback after create one prototype model. And component based rad model which use available component to develop new system

3. Factors affecting the Choice of Software Life Cycle Models in the Software Industry [4]

Table 1: Factors affecting the choice of Software

Factor name
Nature of Project
Project size
Project duration
Project complexity
Level and type of expected risk
Level of understanding of user requirement
Level of understanding of the application area
Customer Involvement
Experience of developers
Team size
Man-machine interaction
Availability of tools and technology
Version of the product
Level of consistency require

3.1 Comparison chart of various model based on usage [4]

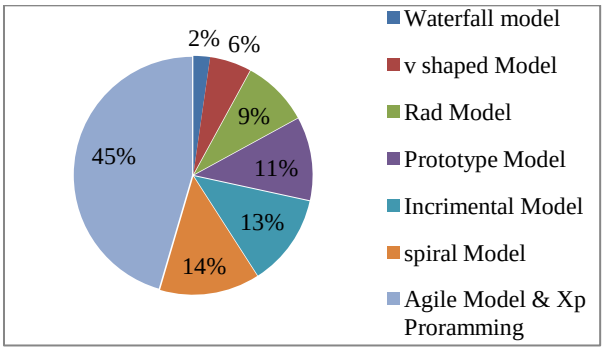

Figure: 2 Comparison Chart of Various process Model Based on usage.

3.2 Comparison of Various Process Model

Various process model for software development are brows here. Table 2 Shows various process modesl which describe in previous section. it Describe which process model satisfied which parameter? Also some of model are highly efficient but not used much and some is much complex but give good Efficiency and used highly in the world of software development.

3. 3 Comparison Table of Various Process Model

Table 2: Comparison of Various Process Model using Different Parameter

Parameter / Process Model→	Waterfall Model	Incremental Model	Prototype Model	Rad Model	Spiral Model	Agile Model	Xp programming
Clear Requirement Specifications	Initial level	Initial level	At medium level	Initial level	Initial level	Change incrementally	Initial level
Feedback from user	No	No	Yes	No	No	No	Yes
Speed to change	Low	High	Medium	No	High	High	High
Predictability	Low	Low	High	Low	Medium	High	High
Risk identification	At initial level	No	No	No	Yes	Yes	Yes
Practically implementation	No	Low	Medium	No	Medium	High	High
Loom	Systematic sequence	Iterative sequence	Priority on customer feedback	Use readymade component	Identification of risk at each stage	Highly customer satisfaction and incremental development[09]	Customer satisfaction and incremental development
Any variation done	Yes-v model	No	No	No	Yes-win win spiral[6]	No	No
Understandability	Simple	Intermediate	Intermediate	Intermediate	Hard	Much complex	Intermediate
Precondition	Requirement clearly defined	Core product should clearly define	Clear idea of Quick Design	Clean idea of Reuse component	No	No	No
Usability	Basic	Medium	High	Medium	Medium	Most use now a days	medium
Customer priority	Nil	Nil	Intermediate	Nil	Intermediate	High	Intermediate
Industry approach	Basic	Basic	Medium	Medium	Medium	High	Medium
Cost	Low	Low	High	very high	Expensive	Much Expensive	High
Resource organization	Yes	Yes	Yes	Yes	No	No	Yes
Elasticity	No	No	Yes	Yes	No	Very high	Medium

4. CONCLUSION

This paper discussed what is software process model and various process models, also compare them with different parameter and highlight the factors for choosing them .paper Present the chart based on usage. However the existing model still can be improve and modified based on less cost, time and high efficient. The developer should find out following Aspects

1. Find out market analysis that why Agile Model is Popular now a days.

2. How can improve efficiency of given model.

5. ACKNOWLEDGMENTS

Thanks to the R.K University Department Of Computer Engineering who have contributed towards development of the Paper.

6. REFERENCES

[1] Research Topics in Software Engineering Unaided Curricular Paradigms as Compute¸ ˜Paradigms of Computation (UCPC)Minho, FEUP July 23, 2009

[2] CTG. MFA – 003, "A Survey of System Development Process Models", Models for Action Project: Developing Practical Approaches to Electronic Records Management and Preservation, Center for Technology in Government University at Albany / Sunny, 1998 .

[3] Steve Easterbrook, "Software Lifecycles", University of Toronto Department of Computer Science, 2001.

[4] Factors Affecting the Choice of Software Life Cycle Models in the Software Industry-An Empirical Study Journal of Computer Science 8 (8): 1253-1262, 2012ISSN 1549-3636 © 2012 Science Publications

[5] A Comparison between Five Models Of Software Engineering IJCSI International Journal of Computer

Science Issues, Vol. 7, Issue 5, September 2010 ISSN (Online): 1694-0814

[6] Efficiency of Spiral Model by applying Genetic Algorithm 1 Sachin Sharma, 2Anupriya Jain, 3Seema Sharma, 4Sonia Duggal 1, 2,3,4Dept. Of Computer Applications, Manav Rachna International University, Faridabad, Haryana, India. I S S N : 2229 - 4333 (P r i n t) | I S S N : 0 9 7 6 - 8 4 9 1 (On l i n e) IJCST Vol. 2, Issue 2, June 2011

[7] Emerging Trends in Software Engineering presented by Roger S. Pressman, Ph.D. R.S. Pressman & Associates, Inc. Boca Raton, Florida USA January, 2009

[8] Analysis of various Software Process Models Ashwini Mujumdar, Gayatri Masiwal, P. M. Chawan / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, May-Jun 2012, pp.2015-2021

[9] Agile Maturity Model (AMM): Software Process Improvement framework for Agile Software Development Practices by Chetankumar Patel Muthu Ramachandran. IJSE Vol.2 No.1 January 2009

[10] Roger Pressman, Software Engineering: A Practitioner's Approach, Sixth Edition, McGraw-Hill Publication.

[11] Sanjana Taya and Shaveta Gupta, Comparative Analysis of Software Development Life Cycle Models, IJCST Vol. 2, Issue 4, Oct . - Dec. 2011.