A Comparative study of Different types of Models in Software Development Life Cycle

Sushma Malik[1], Charul Nigam[2]

*[1] Assistant Professor, Dept. Of IT Institute of Innovation in Management & Technology
Janakpuri, New Delhi, India
[2] Assistant Professor, Dept. Of IT Institute of Innovation in Management & Technology
Janakpuri, New Delhi, India*

Abstract - *A software process, also known as a software development life-cycle (SDLC), is a structure imposed on the development of a software product. Software Processes are the lifeline of any Software Development Model. Software Processes decide the survival of a particular software development model in the market as well as in software organization. The set of processes those proved to be effective and efficient for software development in one organization may or may not be followed in another organization. That is other organization finds another approach for software development more convenient to work with.*

The development lifecycle of software Comprises of four major stages namely Requirement Elicitation, Designing, Coding and Testing. A software process model is the basic framework which gives a workflow from one stage to the next. This workflow is a guideline for successful planning, organization and final execution of the software project.

In this research paper the comparison of various software development models has been carried out. According SDLC each and every model have the advantage and drawbacks so in this research we have to calculate the performance of each model on behalf of some important features.

***Key Words*: SDLC (software development life-cycle), SRS (software requirement specification), URD (User Requirement Documentation), SRD (System requirement documentation), Software Process.**

1. INTRODUCTION

Today, the computer has become a very vital part of our life. It is used in various fields of life, such as, industry, medicine, education, commerce and even agriculture. It has become an important element for the Organizations because they depends on computer for their works .Computer is a time-saving gadget and its progress helps in executing complex, long, repeated work in a very short time with a high speed. The number of companies that produce software programs for the purpose of facilitating works of offices, administrations, banks, etc, has increased recently and they follow the some path to accomplished their task.

All software developed by many people, should be developed using some kind of methodology. Even small pieces of software developed by one person can be enhanced by keeping a methodology in mind. A methodology is a systematic way of doing things. It is a repeatable process that we can follow from the earliest stages of software development through to the maintenance of an installed system.

Software development organizations follow some software development process when developing a software product. The primary function of software development process models is to ―determine the order of the stages involved in software development and evolution and to establish the transition criteria for progressing from one stage to the next. In the software development process, the problem (Software) can be dividing in the following activities:

i. Understanding the problem

ii. Decide a plan for the solution

iii. Coding for the designed solution

iv. Testing the definite program

These activities may be very complex for large systems. So, each of the activity has to be broken into smaller sub-activities or steps. These steps are then handled effectively to produce a software project or system. The basic steps involved in software project development are:

1. Requirement analysis
2. Design
3. Coding
4. Testing
5. Maintenance

2. GENERAL APPROACHES

The various approaches which were used to developing a software development process model are as follows:

1) Sequential process model
2) Incremental process model
3) Evolutionary process model

2.1 Sequential Process Model

Sequential approaches (e.g. waterfall model) refer to the completion of the work within one linear cycle. Projects are arranged into a set of steps that are completed serially .Progress is carried out in linear fashion enabling the passing of control and information to the next phase when pre-defined objectives are reached and accomplished.

The Waterfall Model: The waterfall model is the classical model of software engineering. This model is one of the oldest models. The pure waterfall lifecycle consists of several non-overlapping stages. The waterfall model serves as a baseline for many other lifecycle models. The steps followed in the waterfall model are:

1) Requirement analysis and specification: the goal of this phase is to understand the exact requirement s of the customer and to document them properly in a natural language contains a description of what the system will do without describing how it will be done. The resultant document is known as software requirement specification (SRS) document.

2) Design: the goal of this phase is to transform the requirements specification into a structure that is suitable for implementation in some programming language. Overall software architecture is defined and detailed design work is performed. This work is documented and known as software design description (SDD) document.

3) Implementation and unit testing: In this phase design is implemented. If the SDD is complete, the implementation or coding phase proceeds smoothly. During testing, the major activities are centered around the examination and modification of the code. Initially, small modules are tested in isolation from the rest of the software product.

4) Integration and System Testing: The objective of unit testing is to determine that each independent module is correctly work. This gives little chance to determine that the interface between modules is also correct and for this reason integration testing is performed. And system testing involves the testing of the entire system, whereas software is a part of the system. This phase is essential to develop confidence in the developers before software is delivered to the customer or released in the market.

5) Operation and Maintenance: Software maintenance is a task that every development group has to face, when the software is delivered to the customer's site, installed and is operational. Software maintenance is a broad activity that includes error correction, enhancement of capabilities and deletion of obsolete.

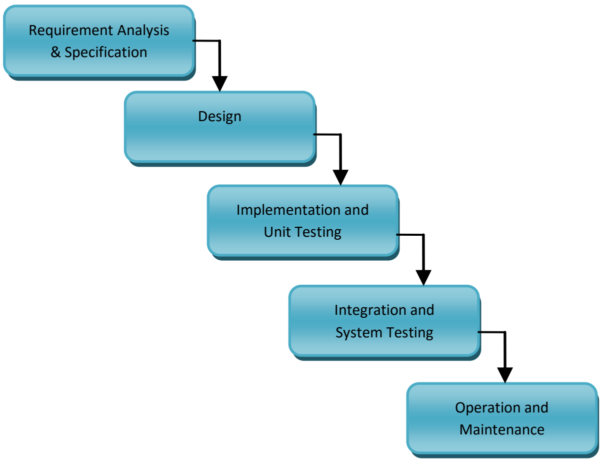
Requirement Analysis & Specification
Design
Implementation and Unit Testing
Integration and System Testing
Operation and Maintenance

Figure -1: Waterfall Model

Advantages:

1. Easy to understand and implement.
2. Widely used and known
3. Reinforces good habits: define-before- design, design-before-code.
4. Identifies deliverables and milestones.
5. Document driven, URD, SRD … etc.
6. Works well on mature products and weak teams.

Disadvantages:

1. Real projects are rarely sequential.
2. It is difficult to define all requirements at the beginning of a project.
3. A working version of the system is not seen until late in the project's life.
4. Software is delivered late in project, delays discovery of serious errors.
5. Difficult to integrate risk management.
6. The biggest disadvantage of the waterfall model is that cannot go back a step; if the design phase has gone wrong, things can get very complicated in the implementation phase.
7. Significant administrative overhead, costly for small teams and projects.

2.2 Incremental Process Model

Increment process models (e.g. Iterative Enhancement model) are effective in the situations where requirements are defined precisely and there is no confusion about the functionality of the final product and functionality can be delivered in phases as per desired priorities. After every cycle, a useable product is given to the customer.

Iterative Enhancement model: This model contains the following phases:

- Requirement analysis and specification
- Design
- Implementation and unit testing
- Integration and System Testing
- Operation and Maintenance

These phases are same as the waterfall model, but these may be conduct in several cycles in Iterative enhancement model. A useable product is released at the end of the each cycle, with each release providing additional functionality. This model does deliver an operational quality product at each release. The complete product is divided into releases and the developer delivers the product release by release. A typical product will usually have many releases. With this model, first release may be available within few weeks or months whereas the customer generally waits months or years to receive a product using other model.

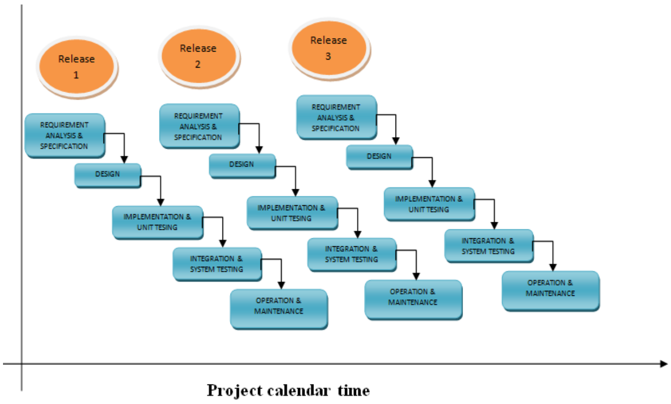
Release 1
Release 2
Release 3
REQUIREMENT ANALYSIS & SPECIFICATION
DESIGN
IMPLEMENTATION & UNIT TESING
INTEGRATION & SYSTEM TESTING
OPERATION & MAINTENANCE
REQUIREMENT ANALYSIS & SPECIFICATION
DESIGN
IMPLEMENTATION & UNIT TESING
INTEGRATION & SYSTEM TESTING
OPERATION & MAINTENANCE
REQUIREMENT ANALYSIS & SPECIFICATION
DESIGN
IMPLEMENTATION & UNIT TESING
INTEGRATION & SYSTEM TESTING
OPERATION & MAINTENANCE
Project calendar time

Figure -2: Iterative Enhancement Model

Advantages:

1. Divides project into smaller parts
2. Creates working model early and provides valuable feedback
3. Feedback from one phase provides design information for the next phase
4. Very useful when more staffing is unavailable

Disadvantages:

1. User community needs to be actively involved in the project. This demands on time of the staff and add project delay
2. Communication and coordination skills take a center stage
3. Informal requests for improvement for each phase may lead to confusion

2.3 Evolutionary Process Model

Evolutionary process model (e.g. Prototyping model) resembles Iterative enhancement model, but this differs from iterative enhancement model in the sense that this does not release product at the end of each cycle. This model is useful for projects using new technology that is not well understood. This is also used for complex projects where all functionality must be delivered at one time, but the requirements are unstable or not well understood at the beginning.

Prototyping model: In this process model, firstly develop a working prototype (i.e., incomplete versions of the software program being developed.) Of the software instead of developing the actual software. The working prototype is developed as per current requirements. The developers use this prototype to refine the requirements and prepare the final specification document. When the prototype is created, it is reviewed by the customer. Typically this review gives feedback to the developer that helps to remove uncertainties in the requirements of the software.

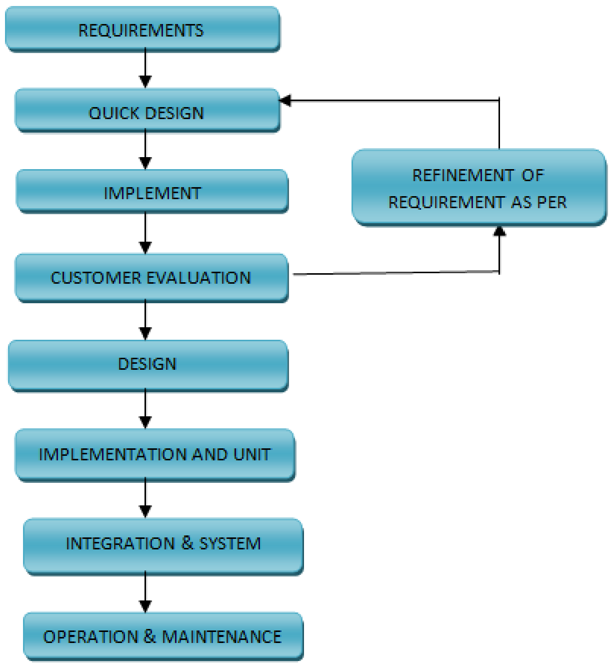

Figure -3: Prototyping Model

Advantages-

1. Early visibility of the prototype gives users an idea of what the final system looks like
   Encourages active participation among users and producer.
2. Enables a higher output for user.
3. Cost effective
4. Increases system development speed
5. Assists to identify any problems with the efficacy of earlier design, requirements analysis and coding activities.

Disadvantages-

1. Possibility of causing systems to be left unfinished.
2. Possibility of implementing systems before they are ready.
3. Producer might produce a system inadequate for overall organization needs.
4. Often lack flexibility.
5. Not suitable for large applications.
6. Project management difficulties.

Table 1. Comparison between different software development models

Model /Features	**Waterfall Model**	**Iterative Enhancement model**	**Prototype model**
Requirement Specifications	Beginning	Beginning	Beginning
Understanding Requirements	Well Understood	Not Well understood	Not Well understood
Cost	Low	Low	High
Simplicity	Simple	Intermediate	Intermediate
Risk Analysis	Only at beginning	No risk analysis	No risk analysis
Flexibility	Rigid	Less Flexible	Flexible
Reusability	Limited	Yes	Yes
User Involvement	Only at beginning	Intermediate	Yes
Complexity of system	simple	complex	complex
Overlapping Phases	No overlapping	No overlapping	overlapping
Implementation time	Long	Less	Less
Guarantee of Success	Less	High	Good
Changes Incorporated	Difficult	Easy	Easy
Expertise Required	High	High	Medium
Resource Control	Yes	Yes	No

3. CONCLUSIONS

There are lots of process models for developing systems and project requirements. Each model has advantages and disadvantages. Each model tries to eliminate the disadvantages of the previous model.

After analysis of waterfall model, Iterative Enhancement Model and Prototype Model models through the various factors, it has been found that the water fall model is used by various big companies for their projects .Since the development team is familiar to the environment and it is feasible to specify all requirements of working environment. Iterative Enhancement model overcome the drawback of waterfall model. It allows feedback to proceeding stage. Prototype model used to develop online systems for transaction processing. Since significantly reduce rework and lead to the creation of working model in lower capital cost.

REFERENCES

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

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

[3] Pankaj Jalote, "An Integrated Approach to Software Engineering", Springer Science Business Media, Inc, Third Edition, 2005.

[4] S. M. Metev and V. P. Veiko, Laser Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998

[5] Jim Hurst, Comparing Software Development Life Cycles