Five Models of Software Development Engineering

Surya Madaan1
1B.tech Student of Computer Science & Engineering
Jaypee Institute of Information Technology
A-10, Sector-62, Noida, Uttar Pradesh 201307, India

Abstract: This paper deals with a vital and important issue in computer Science world. It is concerned with the software development and management processes that examine the area of software development through the development models, which are known as software development life cycle. It represents five of the development models namely, waterfall, Iteration, V-shaped, spiral and Extreme programming. These models have advantages and disadvantages as well. Therefore, the main objective of this research is to represent different models of software development and to understand and show the features and defects of each model.

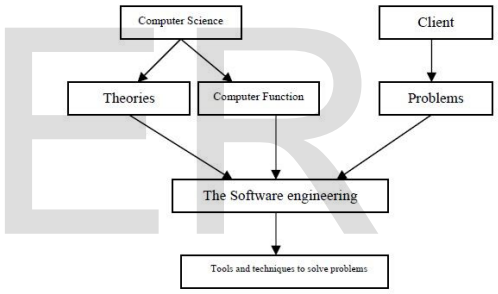
Keywords: Software Development Models, Software Management Processes, Comparison between five models of Software Engineering.

1. Introduction

In are day to day life's computer is everywhere. In fact, computer has become indispensible in today's world as it is used in many fields of life such as industry, medicine, commerce, education and even agriculture. Now days, organizations become more Dependent on computer in their works as a result of computer technology.[1]Computer is considered a time- saving device and its progress helps in executing complex, long, repeated processes in a very short time with a high speed. In addition to using computer for work, people use it for fun and entertainment. Noticeably, the number of companies that produce software programs for the purpose of facilitating works of offices, administrations, banks, etc., has increased recently which results in the difficulty of enumerating such companies. No one can deny the importance of computer in our life, especially during the present time. In fact, computer has become indispensible in today's life. Moreover, Computer science is the scientific and practical approach to computation along with it the aim of software engineering is to create a suitable work that constructs programs of high quality.

.

Fig. 1 Explanation of Software Engineering Conception



2. Software Process Models

A Process Model describes the sequence of phases for the entire lifetime of a product. Therefore it is sometimes also called Product Life Cycle. This covers everything from the initial commercial idea until the final de-installation or disassembling of the product after its use.
It presents a description of a process from some particular perspective as:
1. Specification.
2. Design.
3. Validation.
4. Evolution.

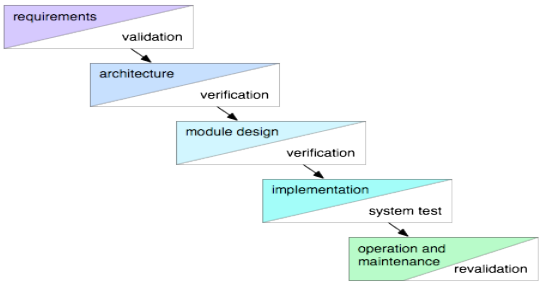


Fig. 2 Software Process Model Diagram

There are many variants of these models e.g. formal development where a waterfall-like process is used, but the specification is formal that is refined through several stages to an implementable design[2].

3. Five Models

A Programming process model is an abstract representation to describe the process from a particular perspective. There are numbers of general models for software processes, like: Waterfall model, Evolutionary development, Formal systems development and Reuse based development, etc. This research will view the following five models :

1. Waterfall model.
2. Iteration model.
3. V-shaped model.
4. Spiral model.
5. Extreme model.

These models are chosen because their features correspond to most software development programs.

3.1 The Waterfall Model

The first known presentation describing use of similar phases in software engineering was held by Herbert D. Benington at Symposium on advanced programming methods for digital computers on 29 June 1956.[3] This presentation was about the development of software for SAGE. In 1983 the paper was republished with a foreword by Benington pointing out that the process was not in fact performed in a strict top-down fashion, but depended on a prototype.[4]

The first formal description of the waterfall model is often cited as a 1970 article by Winston W. Royce,[5][6]

The **waterfall model** is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction and maintenance.

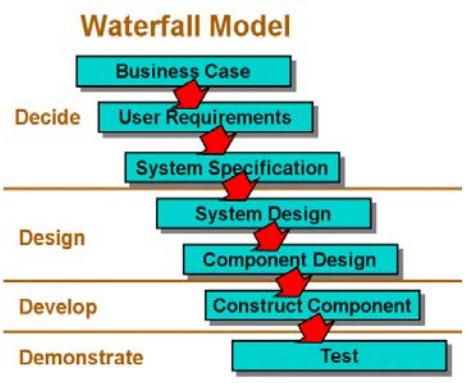


Fig.3 Waterfall Model

Advantages

- Minimizes planning overhead since it can be done up front.
- Structure minimizes wasted effort, so it works well for technically weak or inexperienced staff.

Disadvantages

- Inflexible
- Only the final phase produces a non documentation deliverable.
- Backing up to address mistakes is difficult.

Waterfall development completes the project-wide work-products of each discipline in one step before moving on to the next discipline in the next step. Business value is delivered all at once, and only at the very end of the project. Backtracking is possible in an iterative approach.

3.2 Iterative Development

The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system. Learning comes from both the development and use of the system, where possible key steps in the process start with a simple implementation of a subset of the software requirements and iteratively enhance the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added.

Iterative and Incremental development is any combination of both iterative design or iterative method and incremental build model for software development. The combination is of long standing [7] and has been widely suggested for large development efforts. For example, the 1985 DOD-STD-2167[8] mentions (in section 4.1.2): "During software development, more than one iteration of the software development cycle may be in progress at the same time." and "This process may be described as an 'evolutionary acquisition' or 'incremental build' approach." The relationship between iterations and increments is determined by the overall software development methodology and software development process. The exact number and nature of the particular incremental builds and what is iterated will be specific to each individual development effort.

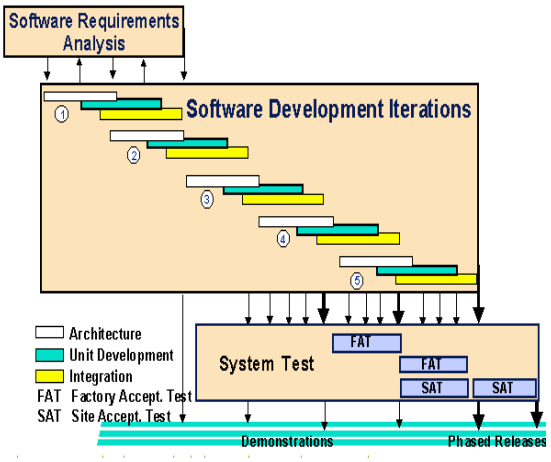


Fig.4 Iterative Development

3.3 V-\\\\\\\\\haped Model

The V-model[11] represents a software development process (also applicable to hardware development) which may be considered an extension of the waterfall model. Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape. The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing. The horizontal and vertical axes represents time or project completeness (left-to-right) and level of abstraction (coarsest-grain abstraction uppermost), respectively.

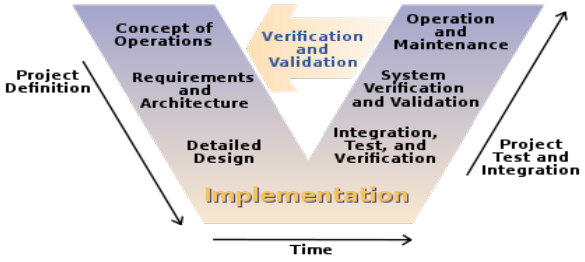


Fig.5 V-Shaped Model

Advantages

- Simple and easy to use.
- Each phase has specific deliverables.
- Higher chance of success over the waterfall model due to the early development of test plans during the life cycle.
- Works well for small projects where requirements are easily understood.

Disadvantages

- Very rigid like the waterfall model.
- Little flexibility and adjusting scope is difficult and expensive.
- Software is developed during the implementation phase,
- so no early prototypes of the software are produced.
- This Model does not provide a clear path for problems found during testing phases [9].

3.4 Spiral Model

This model was first described by Barry Boehm in his 1986 paper "A Spiral Model of Software Development and Enhancement".[12]

The spiral model is a risk-driven process model generator for software projects. Based on the unique risk patterns of a given project, the spiral model guides a team to adopt elements of one or more process models, such as incremental, waterfall, or evolutionary prototyping.

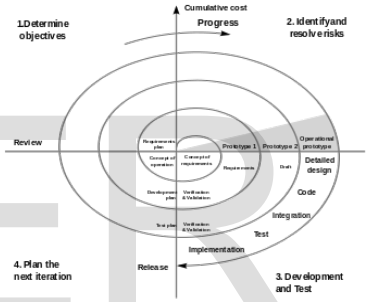


Fig.6 Spiral Model

In the spiral model, the angular component represents progress, and the radius of the spiral represents cost.

Advantages

- High amount of risk analysis.
- Good for large and mission-critical projects.
- Software is produced early in the software life cycle.

Disadvantages

- Can be a costly model to use.
- Risk analysis requires highly specific expertise.
- Project's success is highly dependent on the risk
- analysis phase.
- Doesn't work well for smaller projects [9]

3.5 Extreme Programming

Extreme Programming was created by Kent Beck during his work on the Chrysler Comprehensive Compensation System (C3) payroll project

Software development in the 1990s was shaped by two major influences: internally, object-oriented programming replaced procedural programming as the programming paradigm favored by some in the industry; externally, the rise of the Internet and the dot-com boom emphasized speed-to-market and company growth as

competitive business factors. Rapidly changing requirements demanded shorter product life-cycles, and were often incompatible with traditional methods of software development.
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.

Advantages

- Lightweight methods suit small-medium size projects.
- Produces good team cohesion.
- Emphasises final product.
- Iterative.
- Test based approach to requirements and quality assurance.

Disadvantages

- Difficult to scale up to large projects where documentation is essential.
- Needs experience and skill if not to degenerate into code-and-fix.
- Programming pairs is costly.
- Test case construction is a difficult and specialized Skill. [10]

4. Conclusion and Future Work

After completing this research, it is concluded that :
1. There are many existing models for developing systems for different sizes of projects and requirements.
2. These models were established between 1970 and 1999.
3. Waterfall model and spiral model are used commonly in developing systems.
4. Each model has advantages and disadvantages for the development of systems , so each model tries to eliminate the disadvantages of the previous model

Finally, some topics can be suggested for future works:
1. Suggesting a model to simulate advantages that are found in different models to software process management.
2. Making a comparison between the suggested model and the previous software processes management models.
3. Applying the suggested model to many projects to ensure of its suitability and documentation to explain its mechanical work.

REFERENCES

[1] www.ijcsi.org/papers/7-5-94-101.pdf
[2] Ian Sommerville, "Software Engineering", Addison Wesley, 7th edition, 2004.
[3] United States, Navy Mathematical Computing Advisory Panel (29 June 1956),*Symposium on advanced programming methods for digital computers*, [Washington, D.C.]: Office of Naval Research, Dept. of the Navy, OCLC 10794738
[4] Benington, Herbert D. (1 October 1983). "Production of Large Computer Programs"(PDF). *IEEE Annals of the History of Computing* (IEEE Educational Activities Department) **5**(4): 350–361. doi:10.1109/MAHC.1983.10102. Retrieved 2011-03-21.
[5] Royce, Winston (1970), "Managing the Development of Large Software Systems"(PDF), *Proceedings of IEEE WESCON* **26** (August): 1–9
[6] Wasserfallmodell > Entstehungskontext, Markus Rerych, Institut für Gestaltungs- und Wirkungsforschung, TU-Wien. Retrieved on 2007-11-28 fromhttp://cartoon.iguw.tuwien.ac.at/fit/fit01/wasserfall/entstehung.html
[7] Larman, Craig (June 2003). "Iterative and Incremental Development: A Brief History" (PDF). *Computer* **36** (6): 47–56.doi:10.1109/MC.2003.1204375. ISSN 0018-9162
[8] DOD-STD-2167 Defense Systems Software Development (04 JUN 1985) on everyspec.com
[9] Rlewallen, "Software Development Life Cycle Models", 2005 ,http://codebeter.com.
[10] Karlm, "Software Lifecycle Models', KTH,2006 .
[11] Kevin Forsberg and Harold Mooz, "The Relationship of System Engineering to the Project Cycle," in Proceedings of the First Annual Symposium of National Council on System Engineering, October 1991: 57–6
[12] Boehm B, "A Spiral Model of Software Development and Enhancement", ACM SIGSOFT Software Engineering Notes, ACM, 11(4):14-24, August 1986