

A Survey of Software Development Process Models in Software Engineering

Iqbal H. Sarker¹, Faisal Faruque¹, Ujjal Hossen² and Atikur Rahman¹

¹Department of Computer Science & Engineering (CSE)

Chittagong University of Engineering & Technology, Bangladesh

Email: {iqbal@cuet.ac.bd, faisal_uits@yahoo.com, atique_30@yahoo.com}

²Department of Computer Science & Engineering (CSE)

Bangladesh University, Dhaka, Bangladesh

ujjal.cse1@gmail.com

Abstract

Software has been a significant part of modern society for a long time. In particular, this paper is concerned with various software development process models. Software process model is a description of the sequence of activities carried out in a software engineering project, and the relative order of these activities. It represents some of the development models namely, waterfall, v-shaped, incremental, RAD, iterative spiral and agile model. Therefore, the main objective of this paper is to represent different models of software development and different aspects of each model to help the developers to select specific model at specific situation depending on customer demand.

Keywords: *Software development, Process models, SDLC, Software engineering*

1. Introduction

Software Engineering is concerned with designing, writing, testing, implementing and maintaining software. It forms the basis of operational design and development to all computer systems. Functionality of computers is because of software. A software development process, also known as a software development life cycle (SDLC), is a structure imposed on the development of a software product. It is often considered as a subset of system development life cycle.

There are several models for such processes, each describing approaches to a variety of activities that take place during the process. The development models are the various processes or methodologies that are being selected for the development of the project depending on the project's aims and goals. There are many development life cycle models that have been developed in order to achieve different required objectives. The models specify the various stages of the process and the order in which they are carried out. The selection of model has very high impact on the testing that is carried out. It will define the what, where and when of our planned testing, influence regression testing and largely determines which test techniques to use.

There are various Software development models or methodologies. They are as follows:

- Waterfall model
- V model
- Incremental model
- RAD model
- Iterative model
- Spiral model

- Agile model

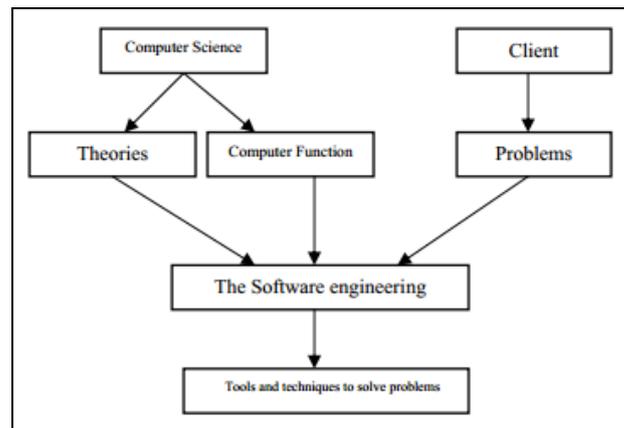


Figure 1. Software Engineering Conception

2. Process Models

In this section, we briefly discuss all the software development process models including the advantages and disadvantages of each model. Moreover, we also mention the different suitable cases of using these models.

A. Waterfall Model

The waterfall model is a sequential design process, often 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, implementation, testing and maintenance. In this model, each phase must be completed fully before the next phase can begin. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. Moreover, in this model phases do not overlap.

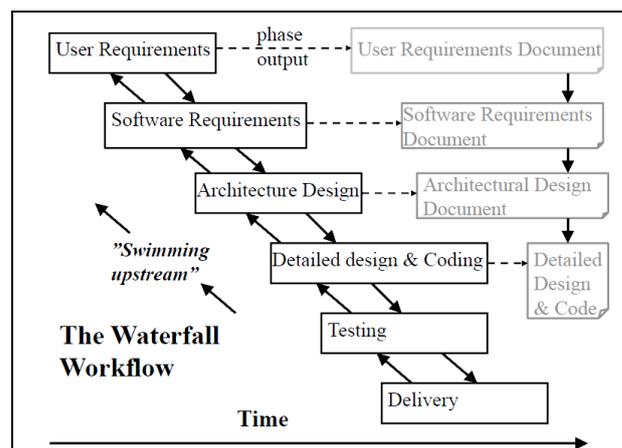


Figure 2. Waterfall Model

We have to use waterfall model in the following cases, such as- 1) When requirements are very well known, clear and fixed, 2) Product definition is stable, 3) Technology is understood, 4) There are no ambiguous requirements, sufficient resources with required expertise are available and 5) The project is short.

Advantages of Waterfall Model:

- Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model, because each phase has specific deliverables and a review process and completed one at a time.
- Works well for smaller projects where requirements are very well understood and sufficient.

Disadvantages of Waterfall Model:

- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- It is not good for long, complex and object-oriented projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.

B. V-Model

V- Model means Verification and Validation model. Just like the waterfall model, the V-Shaped life cycle is a sequential path of execution of processes. Each phase must be completed before the next phase begins. Testing of the product is planned in parallel with a corresponding phase of development.

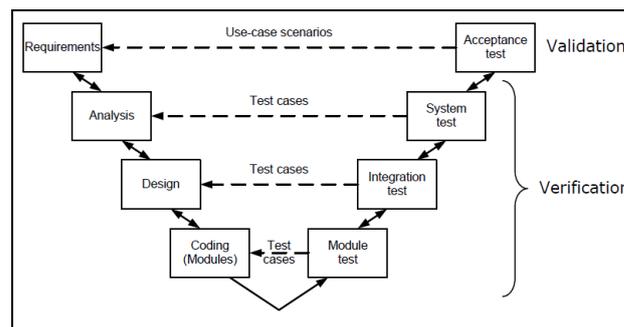


Figure 3. V-model

We have to use V- model in the following cases, such as-

1) The V-shaped model should be used for small to medium sized projects where requirements are clearly defined and fixed, 2) The V-Shaped model should be chosen when ample technical resources are available with needed technical expertise.

Advantages of V-model:

- Simple and easy to use.
- Testing activities like planning, test designing happens well before coding. This saves a lot of time. Hence higher chance of success over the waterfall model.
- Proactive defect tracking – that is defects are found at early stage.
- Avoids the downward flow of the defects.
- Works well for small projects where requirements are easily understood.

Disadvantages of V-model:

- Very rigid and least flexible.
- Software is developed during the implementation phase, so no early prototypes of the software are produced.

- If any changes happen in midway, then the test documents along with requirement documents has to be updated.

C. Incremental Model

In incremental model the whole requirement is divided into various builds. Multiple development cycles take place here, making the life cycle a “multi-waterfall” cycle. Cycles are divided up into smaller, more easily managed modules. Each module passes through the requirements, design, implementation and testing phase. A working version of software is produced during the first module, so you have working software early on during the software life cycle. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is achieved.

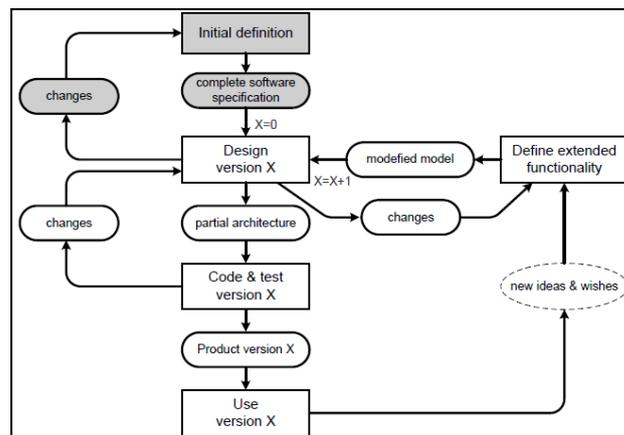


Figure 4. Incremental Model

We have to use incremental model in the following cases, such as- 1) Requirements of the complete system are clearly defined and understood, 2) Major requirements must be defined; however, some details can evolve with time, 3) There is a need to get a product to the market early, 4) A new technology is being used, 5) Resources with needed skill set are not available, 6) There are some high risk features and goals.

Advantages of Incremental Model:

- Generates working software quickly and early during the software life cycle.
- More flexible – less costly to change scope and requirements.
- Easier to test and debug during a smaller iteration.
- Customer can respond to each built.
- Lowers initial delivery cost.
- Easier to manage risk because risky pieces are identified and handled during its iteration.

Disadvantages of Incremental Model:

- Needs good planning and design.
- Needs a clear and complete definition of the whole system before it can be broken down and built incrementally.
- Total cost is higher than waterfall.

D. RAD Model

RAD model is Rapid Application Development model. It is a type of incremental model. In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled 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.

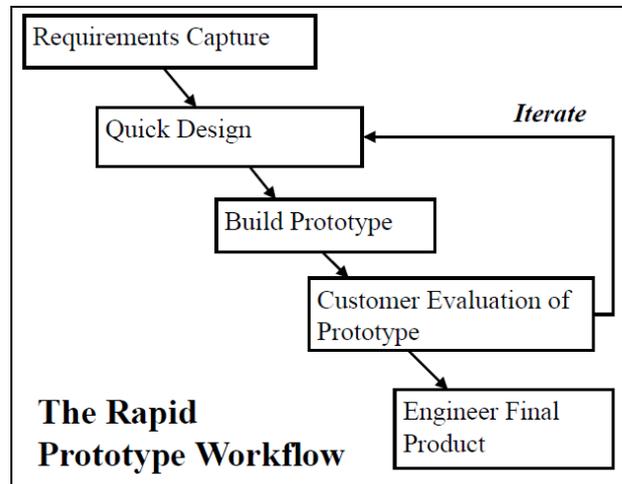


Figure 5. RAD Model

We have to use RAD model in the following cases, such as- 1) RAD should be used when there is a need to create a system that can be modularized in 2-3 months of time. 2) It should be used if there's high availability of designers for modeling and the budget is high enough to afford their cost along with the cost of automated code generating tools. 3) RAD SDLC model should be chosen only if resources with high business knowledge are available and there is a need to produce the system in a short span of time (2-3 months).

Advantages of the RAD Model:

- Reduced development time.
- Increases reusability of components
- Quick initial reviews occur
- Encourages customer feedback
- Integration from very beginning solves a lot of integration issues.

Disadvantages of RAD Model:

- Depends on strong team and individual performances for identifying business requirements.
- Only system that can be modularized can be built using RAD
- Requires highly skilled developers/designers.
- High dependency on modeling skills
- Inapplicable to cheaper projects as cost of modeling and automated code generation is very high.

E. Iterative Model

An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This

process is then repeated, producing a new version of the software for each cycle of the model.

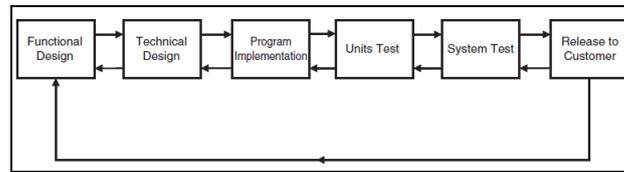


Figure 6. Iterative Model

We use Iterative model in these situations.

1) Requirements of the complete system are clearly defined and understood, 2) When the project is big. 3) Major requirements must be defined; however, some details can evolve with time.

Advantages of Iterative Model:

- In iterative model we can only create a high-level design of the application before we actually begin to build the product and define the design solution for the entire product. Later on we can design and built a skeleton version of that, and then evolved the design based on what had been built.
- In iterative model we are building and improving the product step by step. Hence we can track the defects at early stages. This avoids the downward flow of the defects.
- In iterative model we can get the reliable user feedback. When presenting sketches and blueprints of the product to users for their feedback, we are effectively asking them to imagine how the product will work.
- In iterative model less time is spent on documenting and more time is given for designing.

Disadvantages of Iterative Model:

- Each phase of an iteration is rigid with no overlaps
- Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire lifecycle

F. Spiral Model

The spiral model is similar to the incremental model, with more emphasis placed on risk analysis. The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation. A software project repeatedly passes through these phases in iterations (called Spirals in this model). The baseline spiral, starting in the planning phase, requirements is gathered and risk is assessed. Each subsequent spiral builds on the baseline spiral. Requirements are gathered during the planning phase. In the risk analysis phase, a process is undertaken to identify risk and alternate solutions. A prototype is produced at the end of the risk analysis phase. Software is produced in the engineering phase, along with testing at the end of the phase. The evaluation phase allows the customer to evaluate the output of the project to date before the project continues to the next spiral.

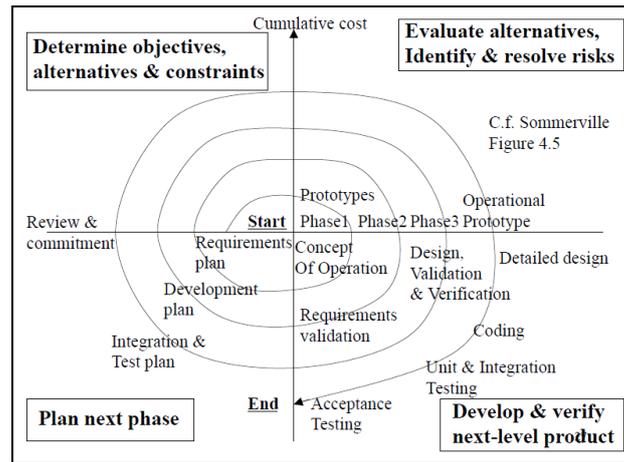


Figure 7. Spiral Model

We have to use spiral model in the following cases, such as- 1) When costs and risk evaluation is important 2) For medium to high-risk projects 3) Long-term project commitment unwise because of potential changes to economic priorities 4) Users are unsure of their needs 5) Requirements are complex 6) New product line 7) Significant changes are expected (research and exploration)

Advantages of Spiral Model:

- High amount of risk analysis hence, avoidance of Risk is enhanced.
- Good for large and mission-critical projects.
- Strong approval and documentation control.
- Additional Functionality can be added at a later date.
- Software is produced early in the software life cycle

Disadvantages of Spiral Model:

- 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.

G. Agile Model

Agile development model is also a type of Incremental model. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly tested to ensure software quality is maintained. It is used for time critical applications. Extreme Programming (XP) is currently one of the most well-known agile development life cycle model.

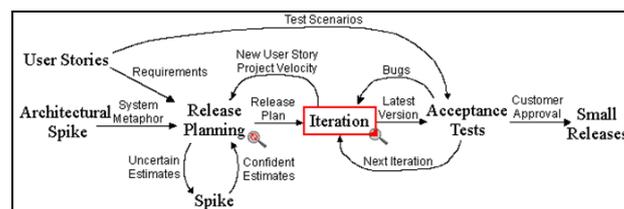


Figure 8. Agile Model

We have to use agile model in the following cases, such as - 1) when new changes are needed to be implemented. The freedom agile gives to change is very important. New changes can be implemented at very little cost because of the frequency of new increments that are produced. 2) To implement a new feature the developers need to lose only the work of a few days, or even only hours, to roll back and implement it. 3) Unlike the waterfall model in agile model very limited planning is required to get started with the project. Agile assumes that the end users' needs are ever changing in a dynamic business and IT world. Changes can be discussed and features can be newly affected or removed based on feedback. This effectively gives the customer the finished system they want or need. 4) Both system developers and stakeholders alike, find they also get more freedom of time and options than if the software was developed in a more rigid sequential way. Having options gives them the ability to leave important decisions until more or better data or even entire hosting programs are available; meaning the project can continue to move forward without fear of reaching a sudden standstill.

Advantages of Agile Model:

- Customer satisfaction by rapid, continuous delivery of useful software.
- People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
- Working software is delivered frequently (weeks rather than months).
- Face-to-face conversation is the best form of communication.
- Close daily cooperation between business people and developers.
- Continuous attention to technical excellence and good design.
- Regular adaptation to changing circumstances.
- Even late changes in requirements are welcomed.

Disadvantages of Agile Model:

- In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
- There is lack of emphasis on necessary designing and documentation.
- The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
- Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

3. Conclusion

A study is given about different development models and their comparison. This paper explained seven different models out of those. First one is Waterfall model which provides base for other development models. Then its enhanced models are explained in Iterative model, Spiral model, V shaped model and finally, Agile development model. The comparison includes the advantages and disadvantages of different models which can help to select specific model at specific situation depending on customer demand.

References

- [1] K. Park, M. Ali and F. Chevalier, "A spiral process model of technological innovation in a developing country: The case of Samsung", African Journal of Business Management, vol. 5, no. 13, (2011), pp. 5162-5178.
- [2] http://en.wikipedia.org/wiki/Software_development_process.
- [3] P. Patil, "Quick Introduction to Agile Software Development", www.indicthreads.com, (2008) June 16.
- [4] S. Easterbrook, "Software Lifecycles", University of Toronto Department of Computer Science, (2001).
- [5] I. Sommerville, "Software Engineering", Addison Wesley, 7th edition, (2004).

- [6] W. S. Humphrey, "A discipline for software engineering", Addison-Wesley Longman Publishing Co., Inc., (1995).
- [7] J. Feller and B. Fitzgerald, "Understanding open source software development", London: Addison-Wesley, (2002).
- [8] T. Stahl, M. Voelter and K. Czarnecki, "Model-driven software development: technology, engineering, management", John Wiley & Sons, (2006).
- [9] I. Jacobson, *et al.*, "The unified software development process", Reading: Addison-wesley, vol. 1, (1999).
- [10] M. Völter, *et al.*, "Model-driven software development: technology, engineering, management", John Wiley & Sons, (2013).

Authors



Iqbal H. Sarker, received his B.Sc. and M.Sc. Engineering in Computer Science & Engineering from Chittagong University of Engineering & Technology (CUET), Bangladesh. Currently, he is serving as an assistant professor in the same department and university. His research interests include software engineering, context-aware mobile computing, and data mining.



Md. Faisal Faruque, received his B. Sc. degree in Computer Science & Engineering from IIUC. Currently, he is pursuing M.Sc. in Computer Science & Engineering in Chittagong University of Engineering & Technology. Now he is serving as an Asst. Professor in Faculty of Engineering at University of Information Technology & Sciences (UITs), Chittagong, Bangladesh. His research interests include Wireless Communication, Security Systems, and Software Engineering.



Md. Ujjal Hossen, is pursuing his B.Sc. in Computer Science & Engineering from Bangladesh University. His research interest is software engineering.



Md. Atikur Rahman, received his B.Sc. in Computer Science & Engineering from Chittagong University of Engineering & Technology (CUET), Bangladesh. Currently, he is serving as Senior Officer in the department of Information Technology (IT) in Meghna Petroleum Limited. His research interest is software engineering.

