

STUDY & COMPARISON OF SOFTWARE DEVELOPMENT LIFE CYCLE MODELS

Gourav Khurana *

Sachin Gupta **

ABSTRACT

This paper deals with a vital and important issue in software engineering world. It is concerned with the software development processes which are known as software development life cycle. There are tons of SDLC models. This paper represents some of those models namely Waterfall model, Iterative model, V-shaped model, Spiral model & Agile model. Each model contains specific activities to be performed to develop software and also have advantages and disadvantages as well. Therefore, the main objective of this paper is to represent different models of software development and make a comparison between them to show the features and defects of each model.

Keywords: *Software Development Life Cycle, SDLC models, Comparison.*

* Haryana College of Technology & Management, Kaithal, India.

** University Institute of Engineering & Technology, Kurukshetra University, Kurukshetra.

1. INTRODUCTION

Software Development Life Cycle (SDLC) methodologies are mechanisms to assure that software meet established requirements. These methodologies impose various degrees of discipline to the software development process with the goal of making the process more efficient and predictable.

In today era, companies have tons of choices of models to adopt for software development. Each model satisfies specific need of customer. For example, company has to adopt RAD model to develop software for the customer demanding his product very soon. So this is a challenging task which model is to select for development. So purpose of this paper is to give a study about some most common development models and their comparison that can help to select a specific SDLC model in particular situation (i.e. according to customer need which model suits best).

2. SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

Software development describes a way, most commonly a sequence of phases or major events and activities that has been found to lead to success in some endeavor. Each software development project has to go through at least the following stages: Software development (Requirement gathering, Writing functional specifications, Creating architecture and design documents, Implementation and coding), Testing and quality assurance, etc. There are various development models such as waterfall model, spiral model, etc. some of these models are explained as below. [4, 5]

3. VARIOUS SDLC MODELS

The various development models are explained as below:

- **Waterfall model:** This is the most common and classic of life cycle models, also referred to as a linear-sequential life cycle model.

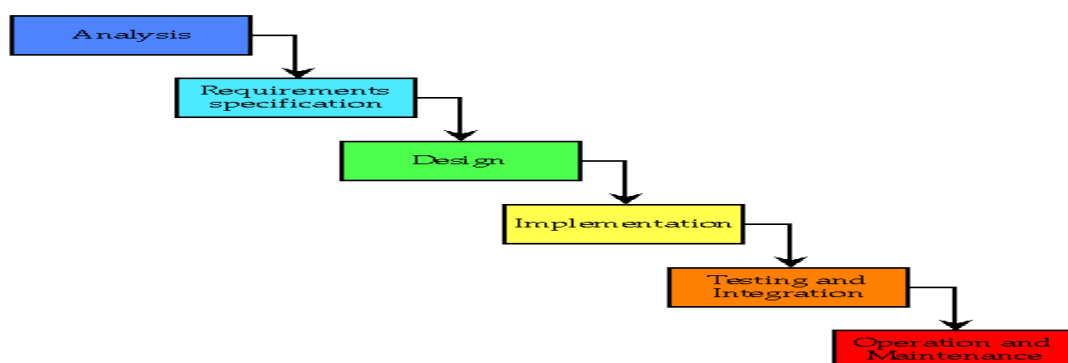


Figure 1: Linear Sequential Model [6]

It is very simple to understand and use. In a waterfall model, each phase must be completed in its entirety 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. [3]

- **Iterative model:** The problems with the Waterfall Model created a demand for a new method of developing systems which could provide faster results, require less up-front information, and offer greater flexibility. With Iterative Development, the project is divided into small parts. This allows the development team to demonstrate results earlier on in the process and obtain valuable feedback from system users. Often, each iteration is actually a mini-Waterfall process with the feedback from one phase providing vital information for the design of the next phase. In a variation of this model, the software products, which are produced at the end of each step (or series of steps), can go into production immediately as incremental releases.

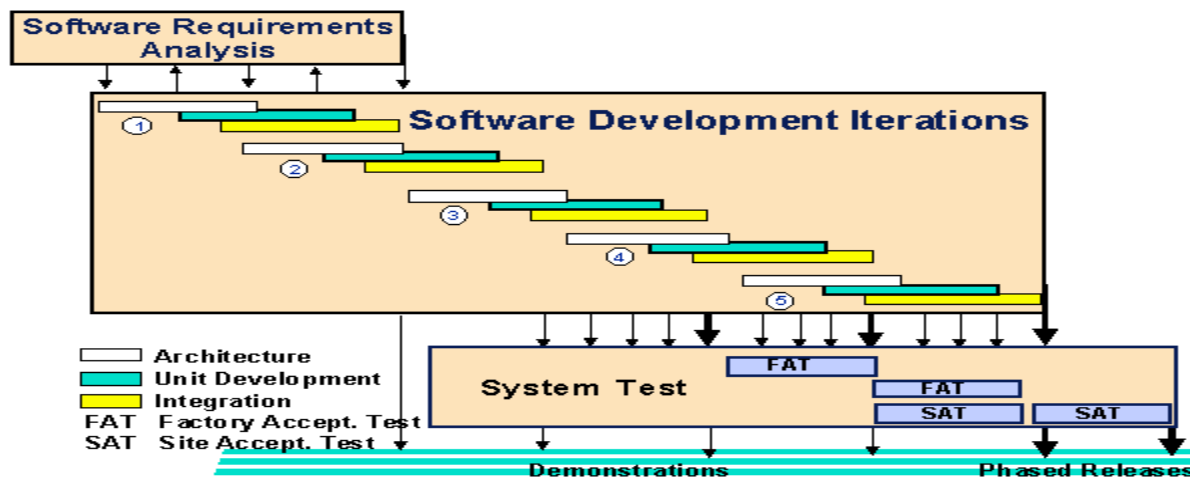


Figure 2: Iterative Model [7]

- **Spiral Model:** The spiral model is similar to the incremental model, with more emphases 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. 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. In the spiral model, the angular component represents progress, and the radius of the spiral represents cost. [1,2]

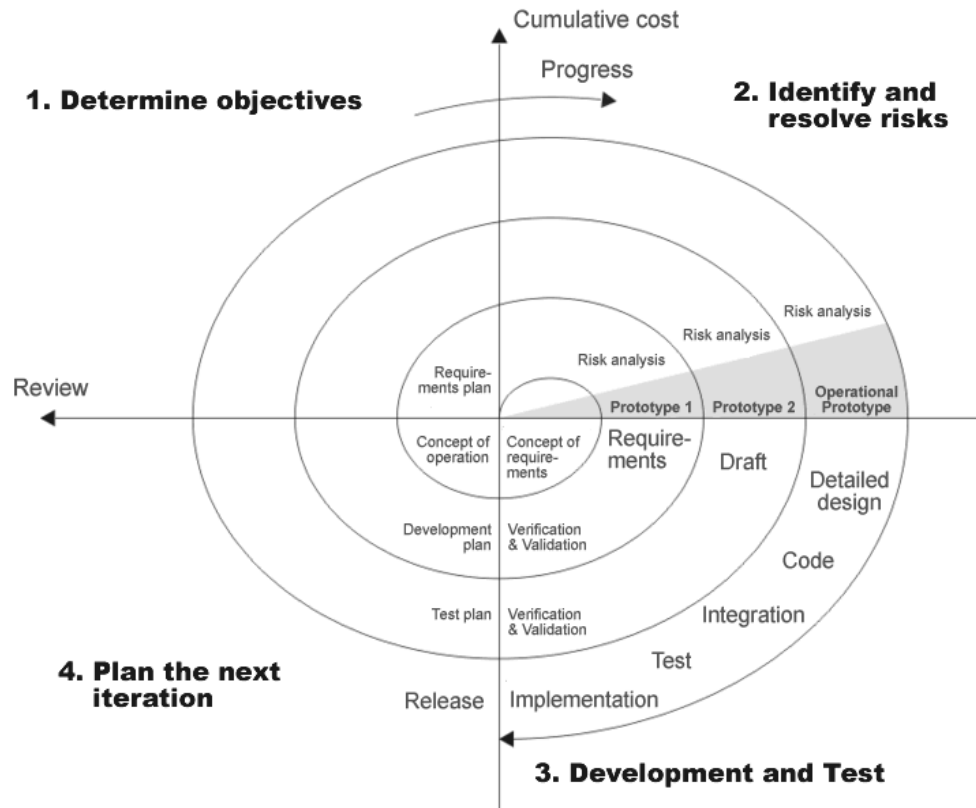


Figure3: Spiral Model [8]

This model of development combines the features of the prototyping and the waterfall model. The spiral model is intended for large, expensive and complicated projects. [8]

- V-Shaped 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 is emphasized in this model more so than the waterfall model though. The testing procedures are developed early in the life cycle before any coding is done, during each of the phases preceding implementation. Requirements begin the life cycle model just like the waterfall model. Before development is started, a system test plan is created. The test plan focuses on meeting the functionality specified in the requirements gathering. The high-level design phase focuses on system architecture and design. An integration test plan is created in this phase as well in order to test the pieces of the software systems ability to work together. The low-level design phase is where the actual software components are designed, and unit tests are created in this phase as well. The implementation phase is, again, where all coding takes place. Once coding is complete,

the path of execution continues up the right side of the V where the test plans developed earlier are now put to use. [3]

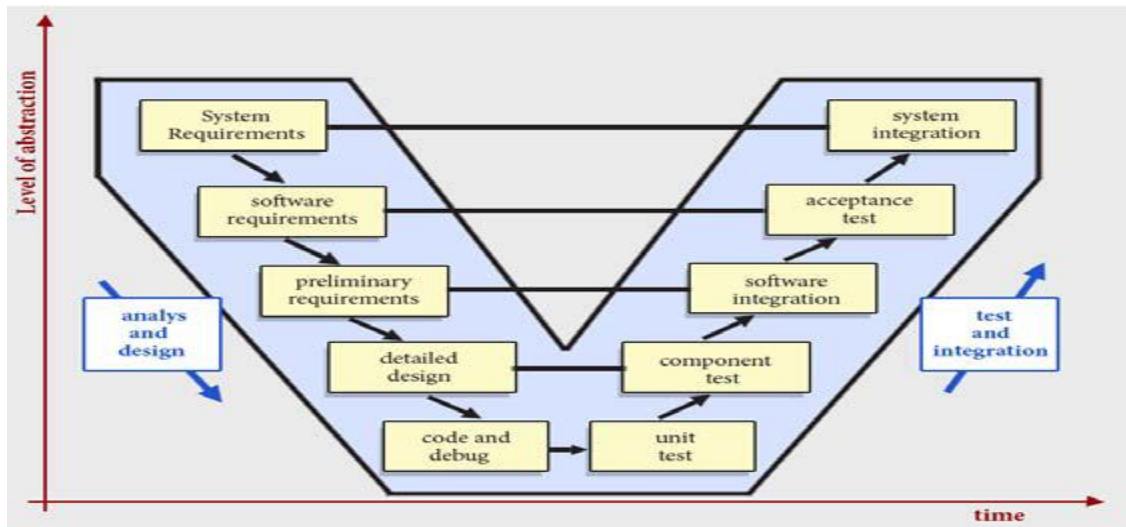


Figure4: V Shaped Model [1]

- **Agile Model:** Agile software development is a style of software development that emphasizes customer satisfaction through continuous delivery of functional software”. The Process of Agile Software Development involves the following:
 1. Starts with a kick-off meeting
 2. The known requirements are understood and prioritized. The development is plan is drawn accordingly.
 3. Relative complexity of each requirement is estimated
 4. Sufficient design using simple diagrams is done
 5. Test Driven Development (TDD) approach may be used. TDD emphasizes on “writing test first and then writing code to pass the test”. It can help in avoiding over-coding.
 6. Development is done, sometimes in pairs, with lot of team interaction. Ownership of code is shared when pair programming is done.
 7. The code is tested more frequently. Sometime a dedicated “Continuous Integration” Server/Software may be used to ease the integration testing of the code.
 8. Depending on the feedback received, the code is refactor. Refactoring does not impact the external behavior of the application but the internal structure may be changed to provide better design, maintainability. Some ways of refactoring may be add interface, use super class, move the class etc.

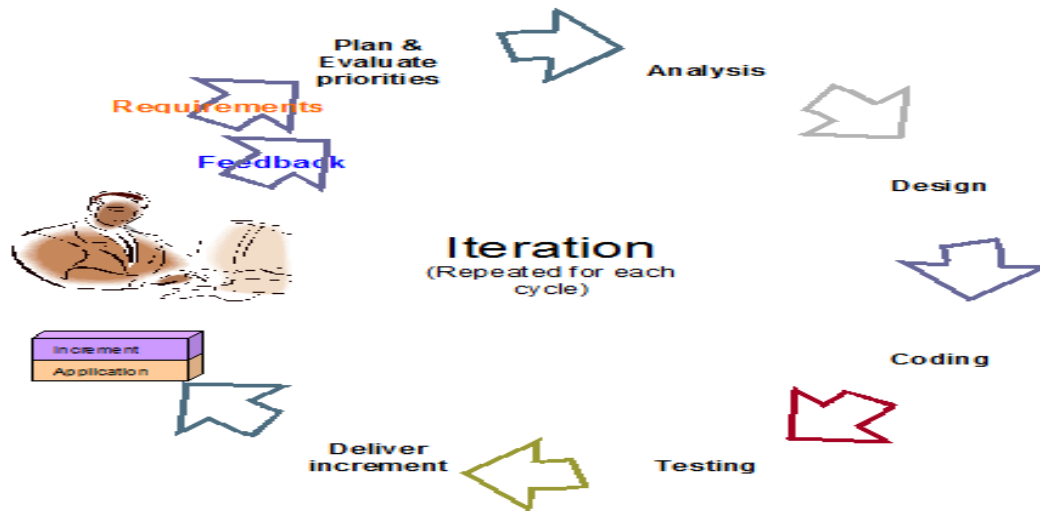


Figure5: Agile SDLC [9]

4. COMPARISON

The comparison between different models is shown by their advantages and disadvantages in two different tables respectively:

S.N	Waterfall model	Iterative model	Spiral model	V shaped Model
1.	Simple and easy to use.	More flexible than the basic waterfall model.	High amount of risk analysis	Simple and easy to use.
2.	Easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.	If there is personnel continuity between the phases, documentation can be substantially reduced.	Good for large and mission-critical projects	Each phase has specific deliverables.
3.	Phases are processed and completed one at a time.	Implementation of easy areas does not need to wait for the hard ones.	Software is produced early in the software life cycle.	Higher chance of success over the waterfall model due to the development of test plans early on during the life cycle.
4.	Works well for smaller projects where requirements are very well understood.	Works well for smaller and moderate size projects	Works well for projects where risk analysis contains higher priority.	Works well for small projects where requirements are easily understood

Table1: Comparison of advantages [3]

S.N	Waterfall model	Iterative model	Spiral model	V shaped Model
1.	Adjusting scope during the life cycle can kill a project.	Milestones are more ambiguous than the waterfall.	Can be a costly model to use.	Very rigid, like the waterfall model.
2.	No working software is produced until late during the life cycle.	Activities performed in parallel are subject to miscommunication and mistaken assumptions.	Risk analysis requires highly specific expertise.	Little flexibility and adjusting scope is difficult and expensive.
3.	High amounts of risk and uncertainty.	Unforeseen interdependencies can create problems.	Project's success is highly dependent on the risk analysis phase.	Software is developed during the implementation phase, so no early prototypes of the software are produced.
4.	Poor model for complex and object-oriented projects. Poor model where requirements are at a moderate to high risk of changing.	Changes are possible as it is iterative model	Doesn't work well for smaller projects.	Model doesn't provide a clear path for problems found during testing phases.

Table2: Comparison of Disadvantages [3]

5. COMPARISON BETWEEN BASIC MODEL (WATERFALL) AND ADVANCED MODEL (AGILE DEVELOPMENT MODEL)

The comparison between Waterfall model (basic model) and Agile model (advanced model i.e. which comes into existence earlier) shows the periodically improvement towards software development life cycle models and it can be shown by following table:

	Waterfall model	Agile Model
History	Waterfall model in software engineering got formally introduced as an idea, through a paper published by Winston Royce in 1970.	Agile model of software development, evolved in the 1990s. After further improvements, In 2001, a group of pioneers in agile software development came together and declared the 'Agile Manifesto', which is a set of canonical rules of sorts, for agile software development methods.
Conceptual Difference	It is a sequential process of software development. Just like in a waterfall, the water progressively falls from one altitude to the lower, in a similar way, the production cycle progresses sequentially, from one stage to the other.	Agile breed of models, focus on 'agility' and 'adaptability' in development. Instead of one time-consuming and rigid development schedule, agile models involve multiple iterative development schedules that seek to improve the output with each iteration.
Efficiency	The ' <i>One Phase</i> ' and ' <i>Rigid</i> ' development cycle of a waterfall model, makes it difficult to make last minute changes in requirements or design.	Agile methods, due to their iterative and adaptable nature, can incorporate changes and release a product, in lesser time.
Suitability	Waterfall model is a natural choice when the customer has provided a clear list of requirements, which are not likely to be modified.	Agile models are applicable in every area of software development. It's best suited for web based applications where its iterative nature helps in incorporating and correcting the various bugs that arise over time.

Table3: Comparison of Waterfall and Agile model [10]

So, this was a waterfall model versus agile model comparison that tried to compare the basic differences between the two. What is ideally needed is a model, which combines the accountability and predictability of a waterfall model, with the agility and adaptability of the agile model. That is, an effective balance between the two ideologies can create a more efficient software development model.

6. CONCLUSION

A study is given about different development models and their comparison. Tons of development models exist. This paper explained five different models out of those. First one is Waterfall model which provides base for other development models. Then its enhanced models are explained. 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.

7. FUTURE SCOPE

The work presented in this paper is survey work of existed software development models including their comparisons.. Nothing in the world which can be said complete. There are a lot of limitations in existing models. In future, a lot of work can be done to produce a new software development model.

8. REFERENCES

1. Steve Easterbrook, "Software Lifecycles", University of Toronto Department of Computer Science, 2001.
2. Kichan Park, Murad Ali and Françoise Chevalier, "A spiral process model of technological innovation in a developing country: The case of Samsung", African Journal of Business Management Vol. 5(13), pp. 5162-5178, 4 July, 2011.
3. Raymond Lewallen, "Software Development Life Cycle", 2005
4. Dale Mayes, "Establishing Great Software Development Process(es) for Your Organization" Class: ETP-410 *Embedded Systems Conference San Francisco*, 2005.
5. Robin F. Goldsmith, "Software development life cycle phases, iterations, explained step by step", Website, URL < <http://searchsoftwarequality.techtarget.com/answer/Software-development-life-cycle-phases-iterations-explained-step-by-step> >, August, 2009
6. <http://www.csse.monash.edu.au/~jonmc/CS2305/Topics/07.13.SWEng1/html/text.html>
7. 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 / Suny, 1998
8. Boehm B, "A Spiral Model of Software Development and Enhancement", ACM SIGSOFT Software Engineering Notes", "ACM", 11(4):14-24, August 1986
9. Prerana Patil, "Quick Introduction to Agile Software Development", www.indicthreads.com, June 16, 2008
10. Gray Pilgrim, "Waterfall Model Vs Agile", Website, <http://www.buzzle.com/articles/waterfall-model-vs-agile.html>, Jan, 2012