

# THE STUDY OF SOFTWARE DEVELOPMENT LIFE CYCLE MODELS

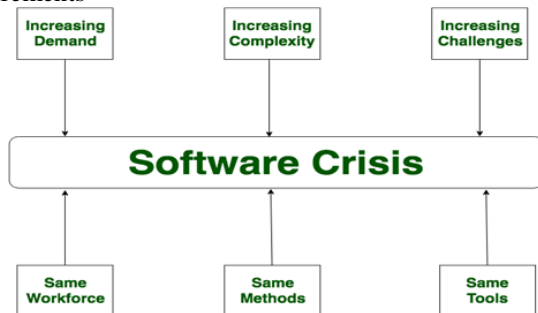
Vanshaj Sharma<sup>1</sup>, Srishti Sharma<sup>2</sup>, Indu Khatri<sup>3</sup>  
<sup>1,2</sup> Students, <sup>3</sup> Professor  
 Mahavir Swami Institute of technology, Sonipat, India

**Abstract:** - Software Development is the becoming the need of every individual. With the advancement of the technology everyone from government to individuals require software to help them into their way. Due to such a demand the competition among the developers arises in the sense of quality and stability. So, for making pre-eminent software's developers use some previously defined Software development models for giving their work a systematic approach and for the risk analysis of the software. Basically, this software development models are the subsets of the software development life cycle (SDLC) models. Each of the Software Development Life model have their own advantages and disadvantages. In this paper we have included seven models of these SDLC models like waterfall model, Rapid Application development model, Spiral model, Iterative Waterfall model, Agile model, Big-Bang model & Prototype model.

**Keywords:** Software Development Life cycle, prototype, Risk Analysis, Software Development models.

## 1. INTRODUCTION

In early day of software development around (1900 to 1969) there was term called software crisis. The Software crisis means difficulty of writing useful and efficient programs. Software crisis causes due to project running over budget, project running over time, software is very inefficient, software is of the low quality and software does not meet the requirements



Solution of these software crisis is software engineering because software engineering provides the systematic approach for the development of the software and can change according to the changing environment. The key concept of the Software engineering is Software Development Life cycle (SDLC) models . SDLC is a descriptive and the diagrammatic representation of the software life cycle.

SDLC models defines entry and the exit criteria for every phase. Without this keeping the track of the progress would be difficult.

There are some key steps of Software development life cycle

models



1. Planning
2. Analysis
3. Design
4. Implementation
5. Maintenance

Software Development life Cycle Models

Waterfall Model

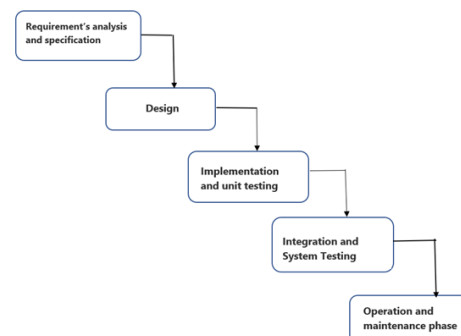
Waterfall model is developed by Winston Royce in 1970. It was developed for making software development process a systematic process so that one who is developing a software would be able

Develop stable software that could be able to run on any environment.

Basically, this model follows an order of phases which cannot be changed and must be followed according to the sequence. The developer must complete every phase before the next phase begins.

This model was named as "Waterfall Model", because its diagrammatic representation resembles with a cascade of waterfalls.

There are basically five phases which needed to be done



Let Discuss about these five phases in deep

1. Requirement's analysis and specification phase: In this phase both developer and customer give their best efforts,

customer give their best in describing what they want from the developer to make and developer give best in understanding what developer have to make. This step is very crucial because this step / phase decide the future of the software.

The requirements could be functions, performance, and interfacing requirement of the software.

In this step we just not only understand them properly but also document these requirements properly in the document called Software Requirement Specification (SRS).

SRS describes the "what" of the system to be produced and not "how."

2. Design Phase: In this phase SRS developed in the previous step transformed into the form from which coding of the program could starts smoothly. Here we make a document called Software Design Document (SDD). In this an architectural design together with high level and detailed design of the software is formed using requirements from SRS and document them in Software Design Document.

3. Implementation and unit testing: In this phase design mentioned in Software Design Document is implemented by the engineers. They use here modular approach of programming (i.e., they implement the design in small modules). After developing small modules, they test these small modules in isolate manner. The process of testing of small modules in isolate manner is called unit testing.

4. Integration and System Testing: In this phase small modules are tested together, to test the interaction between them the system. Because the quality of the end product is determined by the effectiveness of the testing carried out.

5. Operation and maintenance phase: In this phase maintenance of software use to be done. But time period of this phase could be the till retirement of the software.

Conditions for using Waterfall model

1. When the requirements are static means not changing by the period of time.
2. Duration of the project is short
3. When the feedback of the customer is not required.

Advantages of waterfall model

1. This model is very simple to implement.
2. Easy to manage.
3. Phases are processed and implemented at once.
4. Complete project release date and cost to make it are pre-determined.
5. Requirements are very well understood and unchanged during the period of the development.
6. The roadmap to develop software is clearly defined.

Disadvantages of waterfall model

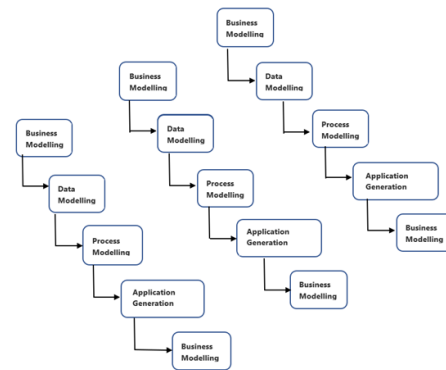
1. In this model risk factor is high so should not be used in complex and significant projects.

2. This model cannot accept the changes in the later stages.
3. There is the difficulty of go back to previous steps.
4. The testing come at the later phases of the model so the risk prediction at the starting is very difficult.

Rapid Application Development Model (RAD)

Rapid Application Development Model is the sequential development process for the software that emphasize on short development cycle using an element based constructive approach. if the requirements are completely determined and clear then time period of the project is fixed. The use of RAD model in software development enables development team to develop fully functional software in short period of time roughly in between 30 to 90 days

Processes of the Rapid Application Development Model are mentioned in flowchart



The various phases of RAD are as follows:

1. Business Modelling: In this phase Developer ask questions about the business for which customer want the software to be ready. These questions are asked to gather the Information flow of their business. Questions are like What data derives the business process, who generates the data, where does the information goes, who process it and so on.

Developer takes the help of Brainstorming activities for gathering that information

2. Data Modelling: In this Information collected in Business modelling are refined into the data object that are needed to support business. The Attributes are identified and the relation between the data objects is identified. These things are done by some experienced team members

3. Process Modelling: In this phase we use to convert the collected data objects in the data modelling phase into to achieve the data flow necessary to implement business. The descriptions are also created for creating, retrieving, updating and deleting data object processes.

4. Application Generation: In this phase construction of the software is being done. But Developers do not use primitive methods to do that. Here Developers not only use Automated

tools for construction of the software but also the 4th generation techniques.

5. Testing & Turnover: In this phase the software developed in the previous phase is tested. But also, with testing the feedback of the customer is taken whether the software is according to him or not. if according to him then it gets finalized else developer again start its process first step again

Note: - Basically, here the division of the software into small modules is done and different teams work on them separately. Each teams perform their task using these above mentioned five steps.

Conditions to use Rapid Application Development model

1. When the complete software is needed in span of 2-3 months.
2. When the requirements are determined.
3. When the risk is limited
4. It should only be used when the budget permits to use the automated tools

Advantages

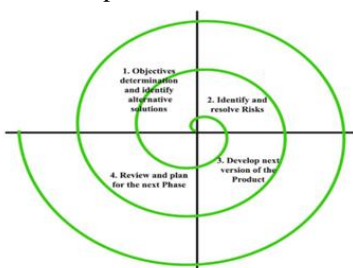
1. The changes can be done here after a module is created.
2. Each phase in Rapid Application Development brings the important functionality related to the customers business for which he wants the software.
3. It reduced the time of development.
4. It supports reusability of the features

Disadvantages

1. Highly skilled developers are required.
2. All the software's are not compatible with the Rapid Application Development model.
3. We cannot use Rapid Application Development model in smaller projects.
4. It is not suitable for highly risk contained projects.
5. Required user involvement.

Spiral Model

Spiral model was developed by the Boehm, it is the evolutionary process in which the iterative features of prototyping is there. Its diagrammatic representation of this model is just like a spring with many loops. It has the potential to develop the new versions of the software rapidly. This model develops the software in the series of incremental releases. During the early iteration, the prototype of software is released. But During the Later iteration more complete versions of software is produced.



The loop in the spiral model is not fixed. The Each loop in

the spiral model represents a phase of a software process.

There are four quadrants of the spiral model

1. Objective setting: In this phase identification of the purpose for each cycle is done. Like feasibility Study and Requirement and Specification.
2. Risk Assessment and reduction: In this we calculate the risk in the project.
3. Development and validation: In this phase strategy making for reducing the risk is done. Strategies are formed using prototype and stimulation.
4. Planning: In this phase planning for the next spiral is done by reviewing the project.

The spiral model is called Meta model because it is better than all other model in the matter of the risk management. But it is more complex than other models available. So, it is recommended in only technically challenging software's.

Conditions for using the Spiral Model

1. When delivery of the software must be frequent.
2. When the project is complex and comparatively large.
3. When customer requirements are unclear and complex.
4. When changes are required to be treated.

Advantages

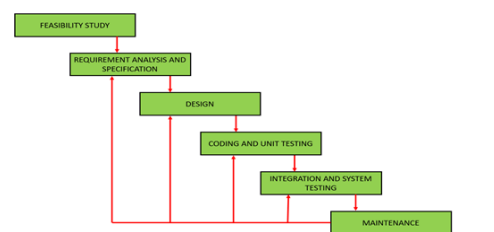
1. Analysis of risk is being done.
2. Useful for large and complex projects.

Disadvantages

1. Budget for the spiral model needs to be high.
2. Expertise is required to analyses the risk.
3. Cannot use in smaller projects.

Iterative Water fall model

The practical application of the waterfall model is not possible because there is not the method to go on to the previous step for any reason. So, for that the iterative waterfall model comes into the play so that it can be useful for practical usage. In this there is a way to give feedback to previous step if current step encounters any error which is the main difference between Waterfall and Iterative Waterfall model.



This feedback path allows giving feedback to previous phase if any error occurs in the current phase due to the mistake of the previous and giving the previous phase a chance for correcting them out. But this feedback path is not available

for feasibility study because once the project is started cannot be stopped very easily.

#### Advantages of the Iterative Water fall model

1. There is the feedback path available for the communication between phases.
2. Iterative waterfall model is very simple to use and understand.
3. It is best suited for agile organizations.
4. In this model more time is for developing and designing than documenting.

#### Disadvantages

1. Cannot handle change request once the requirements in documented.
2. Incremental delivery is not possible for the Iterative waterfall model. Customers have to wait a long time for the complete product.
3. One phase can start after the completion of the previous phase only.
4. Cannot handle risk.

#### Big Bang Model

This model is the simplest model in its form. It requires a little planning, lots of programming and lots of funds. For this model, very small amount of planning is required. It does not follow any process, or at times the customer is not sure about the requirements and the future needs. So the input requirements are arbitrary. This model is conceptualized around the big bang of universe as if we put together lots of programming and funds, we may achieve the best software product. This model works best for small projects with smaller size development team which are working together. It is also useful for academic software development projects and experimenting.

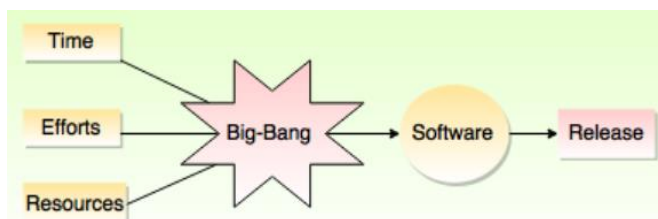


Fig. - Big-Bang Model

<https://cdn.educba.com/academy/wp-content/uploads/2018/12/Big-Bang-Model.png>

#### Advantages:

It is very simple and easy to implement.  
It requires very little or no planning.  
There is no formal procedure are required before starting of any project so easy to handel.

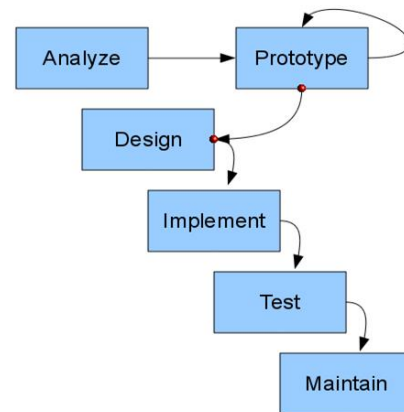
#### Disadvantages:

Very high risk and uncertainty.  
Not good for long and ongoing projects.

#### Prototype Model

This model is one of the most well-known used Software Development Life Cycle Models. This model is used when customers do not know the exact requirement beforehand. A

'quick design' is then created. This design focuses on those aspects of the software that will be visible to the user. It then leads to the development of a prototype. .This is also known as Dummy model. In this process model, the system is partially implemented before or during the analysis phase thereby giving the customers an opportunity to see the product early in the life cycle. The process starts by interviewing the customers and developing the incomplete high-level paper model. This document is used to build the initial prototype supporting only the basic functionality as desired by the customer. Once the customer figures out the problems, the prototype is further refined to eliminate them. The process continues until the user approves the prototype and finds the working model to be satisfactory.



#### Advantages:

1. This model is flexible in design, easy to detect errors.
2. Can find missing functionality easily.
3. Good scope of refinement, new requirements can be easily accommodated.
4. Higher user involvement, enhance reusability.

#### Disadvantages:

1. This model is costly.
2. It has poor documentation because of continuously changing customer requirements.
3. Customers sometimes demand the actual product to be delivered soon after seeing an early prototype.

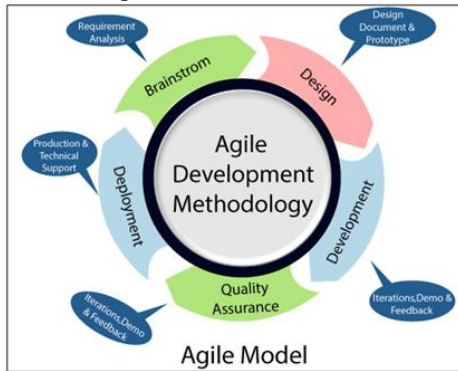
#### Agile Model

This model is recently using by many companies. Agile methodology is a practice which promotes continues interaction of development and testing during the SDLC process of any project. In the Agile method, the entire project is divided into small incremental builds. All of these builds are provided in iterations, and each iteration lasts from one to three weeks

Every iteration involves cross functional teams working simultaneously on various areas like: -

- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing and
- Acceptance Testing.

At the end of the iteration, a working product is displayed to the customer and important stakeholders.



<https://www.tutorialandexample.com/wp-content/uploads/2019/12/Agile-Model-in-Software-Engineering.png>

**Advantages:**

1. This model is most Advanced and Flexible.
2. Customer satisfaction by rapid, continuous delivery of useful software.
3. People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
4. Regular adaptation to changing circumstances.
5. Even late changes in requirements are welcomed.

**Disadvantages:**

1. 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.
2. There is lack of emphasis on necessary designing and documentation.
3. 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.

**Comparative Analysis of seven models**

Models	Waterfall	RAD	spiral	Iterative	agile	Big-Bang	Prototype
Success rate	Low	Better	Better	Good	Best	Average	Good
Time	In years	2-3 months	In years	In years	1 to 4 weeks	In years	In years
Risk analysis	no	High	High	low	High to low	High	Low
Cost	Highest	High	High	Average	Low	High	High
Expertise	Low	High	Average	Low	High	High	High
Customer Involvement	Lowest	High	High	Low	High	High	High
Requirement Handling	Lowest	High	High	Low	High	Low	High

**CONCLUSION**

After completing the study and comparison we conclude that the models are developed from 1970 to 2000. Every model

has its own advantages and disadvantages and every model come into action to overcome the drawbacks of existing model of that time. Now a day's Agile Model is the most latest and commonly used in software development process as it is flexible, advanced, parallel, process divided into sprints by which it takes less time and more user interactive which is need of the hour and other models are used accordingly to the requirement of the software process. All the developers and customers look at low cost, risk, high quality and small cycle of time so that the efficiency and quality of the software can be optimized.

**REFERENCES**

1. <https://www.javatpoint.com/software-engineering-software-development-life-cycle>
2. <https://www.geeksforgeeks.org/software-engineering/>
3. [https://en.wikipedia.org/wiki/Software\\_engineering](https://en.wikipedia.org/wiki/Software_engineering)
4. K.K. Aggarwal & Yogesh Singh, "Software Engineering", New Age International, 2001
5. R. Fairley, "Software Engineering Concepts", Tata McGraw Hill, 1997