



# Comparative Analysis of Software Process Models in Software Development

Sajid Ahmed Ghanghro<sup>1</sup>, Dr. Muhammad Ajmal Sawand<sup>2</sup> Wajid Ahmed Channa<sup>3</sup>, Ubaidulla alias Kashif<sup>4</sup>, Muhammad Hanif Tunio<sup>5</sup>, Kishor Kumar<sup>6</sup>, Nooruddin<sup>7</sup>

<sup>1</sup>Shah Abdul Latif University Khairpur, Pakistan, sajid.ghanghro@salu.edu.pk

<sup>2</sup>Sukkur IBA University Sukkur, Pakistan, ajmal.sawand@iba-suk.edu.pk

<sup>3</sup>Sukkur IBA University Sukkur, Pakistan, wajidahmed.msse14@iba-suk.edu.pk

<sup>4</sup>The Shaikh Ayaz University Shikarpur, Pakistan, kashif.ubaidullah@gmail.com

<sup>5</sup>Shah Abdul Latif University Khairpur, Pakistan, hanif.tunio@salu.edu.pk

<sup>6</sup>Sukkur IBA University Sukkur, Pakistan, kishor.mscss19@iba-suk.edu.pk

<sup>7</sup>Tianjin University, China, noor.cs@yahoo.com

## ABSTRACT

The success and failure of software development depends on the software process models. These models are the key factors in complete development process. It is comprises of various stages i.e., requirement gathering, designing, development, testing and implementation. The contribution of this paper is a comprehensive survey of widely used process models. Moreover, it would definitely play as a guiding path for the researchers to explore future directions for possible research. Extensive analysis of all these five process models is also discussed. Each software process model possesses different factors and parameters which are concisely included so that researchers could easily choose a process model based on the need, resources, and expertise.

**Key words:** Software Process Models; Development, Software Development Life Cycle (SDLC)

## 1. INTRODUCTION

Quality of software development leads to the efficient software process model, also known as “Software Development Life Cycle (SDLC).” In this research paper, we present the comparative analysis of different software process models. Different models are available for different software processes. Each model describes the variety of actions that are performed during the process. “Software process models” are used for the development of projects depending on the nature of the project. We have developed software process models in the decades to improve the quality of “software development.” A “software development process model” is a skeleton proposed for the elaboration of a software product. “Software development processes” consist of different activities, which are the following.

- Requirement Gathering and Analysis
- Software Architecture
- Software Design

- Software Development
- Testing
- Implementation
- Documentation and Training
- Software Maintenance

Software development team considers the aims and goals of a particular project for choosing the software process model to carry out the whole project. There are different types of software development process models and organizations accept the best-suited model which simplifies the “software development process” and increases the throughput of its team members. This research discusses the comparison between five “process models.”

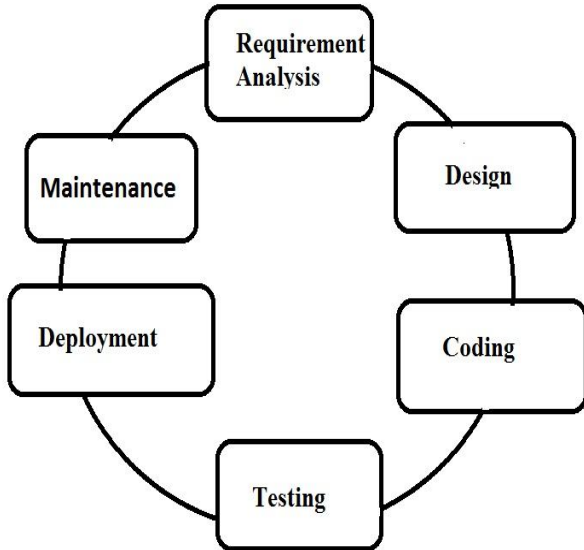
- Waterfall Model
- Iterative Model
- Spiral Model
- Prototype Model
- Agile Model

This research starts with methodology of each listed models. Moreover, it starts with necessary steps in software development life cycle. The comparative analysis of these software process models is also carried out in this research paper.

## 2. SOFTWARE DEVELOPMENT LIFE CYCLE

In this paper, we have examine and compare the main characteristics of OS e.g architecture, scheduling, algorithm, memory protection and management, protocols, real-time and non-real time applications, resource allocation for WSN [5]. In this paper design approaches taken by OS and their weaknesses and supports are also discussed comparatively. In this paper we have compared only OS that have been designed for nodes that have limited resources. Powerful nodes can run more powerful OS and they act as cluster heads/sinks in a WSN [6]. In this paper we will be discussing WSN major design concerns for OS. Then in next section we will be

talking about the importance and need of OS in WSN. Then in the preceding sections you will find the review of popular and trending OS for WSN such as TinyOS, MANSTIS OS, Nano-RK and liteOS [7]. Later in next section comparative analysis and advanced research expected to be done in future is discussed and finally conclusion of this paper.



**Figure 1: Software Development Life Cycle**

**SDLC Phases**

- Requirement gathering and analysis
- Design of System
- Implementation or coding
- Integration and Testing
- Deployment of system
- Maintenance

**3. SOFTWARE PROCESS MODELS**

A “software process model” is abstract representation of the process which are used to develop the software, it only follows the SDLC which includes Design, Implementation, Testing, and Maintenance.

**A. Waterfall Model**

Waterfall model is a traditional model also known as an initial or classical model for software development. Waterfall model describes the development methodology that is linear and consecutive. Due to the subsequent flow of phases in a waterfall model, it cannot go back to its previous stage. After completing a development phase, the process moves to the next step, and cannot turn back to its previous stage. Waterfall model has different aims and goals for each level of its models. It is widely used by Government projects and mature organizations and mostly applicable in case of small projects where there is no chance of changing requirements. In the waterfall model planning is the early stage which helps to

carry the whole project, and it ensures designing flaws we start to develop software. The model begins with collecting the requirement of the system and continues to system and architecture design, coding, testing, deployment, and maintenance. Figure 2 describes the general view of the waterfall model

**Basic Principle of Waterfall Model**

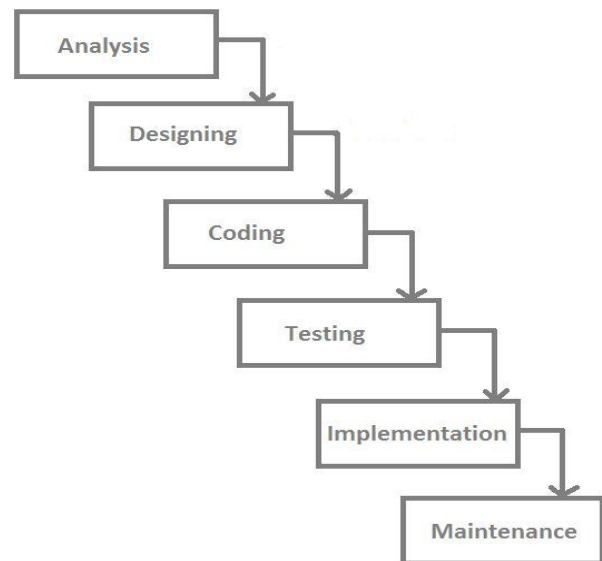
- Project is divided in sequential phases
- Importance should be given to planning of the complete project
- Time schedules for the entire project are defined
- Deadlines should be clearly defined, meeting them is equally important
- Approval of user and management is required at the completion of phase before going to start a new phase.

**Advantages of Waterfall Model**

- Easy to use and understand
- Each step has well-defined milestones
- Work well on mature products and week team
- It acts as like a template in which methods for analysis design, coding and maintenance can place [1]

**Disadvantages of Waterfall Model**

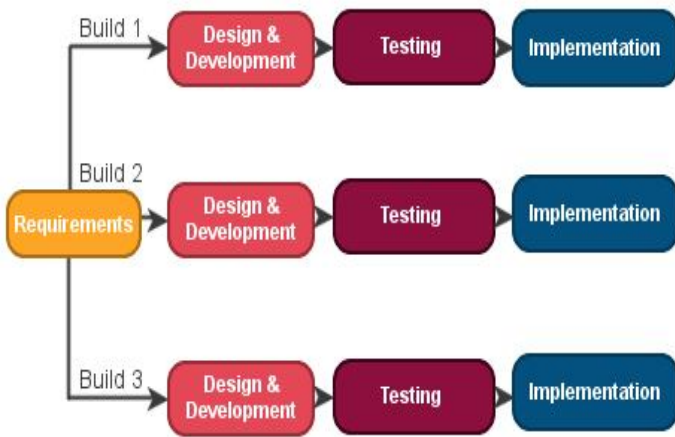
- Requirements are not changeable once designing phase has been completed
- High amount risk and uncertainty
- If the client is not clear at the time of requirement phase, it is tough to accept any change at any stage
- Another significant disadvantage of waterfall model is that a client cannot have any useful product till the final stage of development is not complete.



**Figure 2: Waterfall Model**

**A. Iterative Model**

Iterative design is a method in software development life cycle which focuses on breaking down the software development into small chunks. In the iterative design, features are developed with repeated cycles. The process is started with iterations and goes through with a simple execution of small activities of “software requirements” and incrementally produces the versions of software till the final version is developed. Unlike waterfall model, in the beginning, requirements do not need to be complete. Since development starts with implementing a small part of the software and continues until end or completion of the software to be deployed. While working with the iterative model, we create a rough product or product piece in a single monotony. We then check the product and improve it in next iteration and the same process continues until the completion of the product. The purpose to work iteratively is to allow greater flexibility for changes. While working iteratively, the team goes through a cycle in which iteration is completely evaluated and the changes needed to deliver the required product to the customer are determined.



**Figure 3: Iterative Model**

**Basic Principle**

- The product is divided into small iterations
- User requirements are managed instead of the tasks. User requirements are based on use cases and nonfunctional requirements [2]
- Manage business goals, due to deadlines and budget
- Start with a simple execution of the subsection of requirements that demonstrates the key aspect of the system [2].

**Advantages of Iterative Model**

- Progress can be measured
- Some working functionality can be developed quickly
- Easy to accept changes and are less costly
- Parallel development of different iterations can be plane

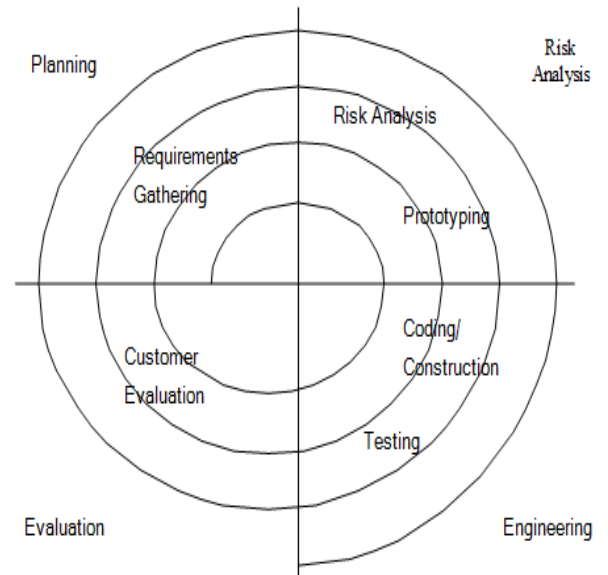
- Testing and debugging is easy due to smaller iterations
- It allows feedback to precede iterations

**Disadvantages of Iterative Model**

- No clear goal is for the iterations
- More resources may be required
- Complex management in the team to manage the iterations
- System and design issue may be rise due to incomplete requirements

**B. Spiral Model**

Spiral model acts as iterative model with more importance on risk inquiry factor in software development. This model is more suitable for large-scale projects. There are specific activities/phases in a single iteration (Spiral) where output is the small prototype of comprehensive software [3]. The activities of the spiral model are “Planning, Analyzing, Engineering, and Evaluation”. A software project frequently passes through these phases in iterations called “Spirals” for the entire duration of software development.



**Fig. Spiral Model**

**Figure 4: Spiral Model**

**Basic Principle of Spiral Model**

- Focus is on risk assessment and minimization of project risks. Project is divided into smaller parts while giving greater flexibility to change during the development process allowing for risk assessment and measuring consideration of the project's continuation throughout its life cycle development [3].
- Each interval around the spiral travels through the activities of spiral model.

**Advantages of Spiral Model**

- High amount of risk analysis
- Estimation of Budget, cost, and Scheduling
- Provide working product at the early stage of SDLC
- Requirements are readily acceptable for changes at any stage

**Disadvantages of Spiral Model**

- Experts are the need for risk factor analysis
- This model can be costly
- Risk of not meeting the schedule or budget
- Project success depends on the risk factor

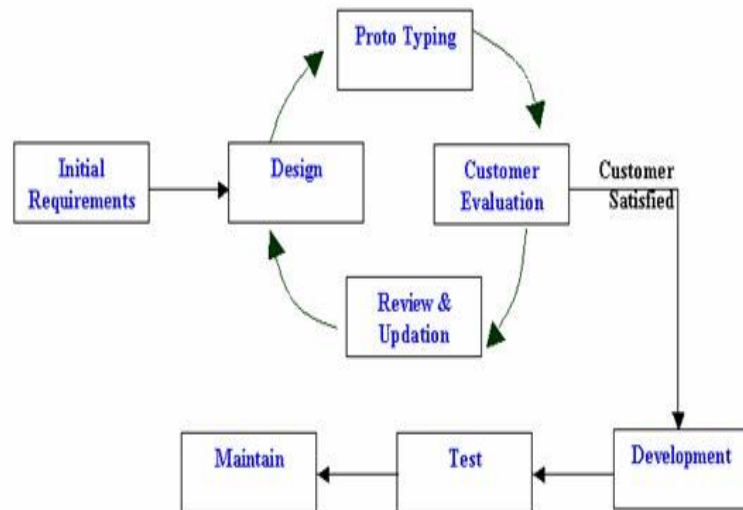
Activity Name	Activities Achieved	Output
Planning	Requirements are Gathered	List of finalized requirements
Risk Analysis	Requirements are considered and stakeholders do the brainstorming to recognize the possible risks	Documents which highlight all the risks and its change procedures
Engineering	Authentic development and testing of software is performed during this phase	- Coding. - Test cases and test results - Test Summary. - Report.
Evaluation	Client assess the software and deliver their response for endorsement or rejection	Features that have been fulfilled are documented

**Table 1: Activities od Spiral Model**

**C. Prototype Model**

The prototype model refers to developing a prototype of a software application which shows the features of the software which is underdeveloped. In prototype model the requirements are not frozen before going to start the design, development and testing phases of the software. The development of the software starts with current known conditions. By this model, the client can get the idea of the system which will be delivered at the end. It helps in obtaining valuable feedback from the customer and helps the software development team to clarify whether the system is accurately acknowledged by the client. The prototype is a working model of software with some limited functions [4]. The prototype does not always keep the exact logic that will be used in the delivered software. It also helps to understand requirements that are unique to the user and may not be taken into account by the developer during the product design.

The prototype is given to the client and the client uses it. Moreover, they provide feedback on the prototype of developers: "What's correct, what needs to be changed, what's missing, what's not needed, etc. Based on the feedback, prototypes are changed to accommodate some of the proposed changes. Users and customers can again therefore be allowed to use the system [4]. This cycle is repeated for prototype assessment. Based on the feedback, basic requirements are changed to produce a final specification, which is then used to develop a quality system.



**Figure 5: Prototype Model**

**Basic Principle of Prototype Model**

- Project risk is reduced by breaking the project into smaller sections and providing flexibility to change during the development process [5]
- The user is involved in the complete development of software, which gives feedback after prototype for the acceptance or any other change in the project
- Small scale mockups are prepared until the user accepts the desired prototype
- While most prototypes are drawn up with some expectations that they will be discarded, it is possible in some cases to evolve from prototype to working system [5].

**Advantages of Prototype Model**

- Prototype gives an idea to the user that how the system will look like finally
- Increases system development speed [6]
- Users have involved actively
- Errors can be detected at early stage
- Missing functionality can be determined easily at first stage [7]
- Helps in reducing the risk factor from the failure of the system

**Disadvantages of Prototype Model**

- This model is time-consuming and expensive in the case until the user accepts the prototype [7]
- The prototype can lead false expectation
- The primary objective of prototypes is rapid development. Thus, system design may seem like it has been developed in series without considering the integration of all other components
- The possibility of causing systems to be left unfinished
- Not suitable for the large type of projects.

**D. Agile Methodology**

Agile development is an incremental model. This method focuses on process flexibility and customer satisfaction by rapid delivery of working “software product.” In Agile, the tasks are broken down into small time boxes which are easy to measure and bring specific features for a release. An iterative approach is carried out, and working features are provided at the end of the each iteration. Iteration is typically of two to four weeks and involves cross-functional team to work simultaneously on the project. In each iteration of an agile process, all activities like planning, designing, and development and testing [11] are performed. In agile methodology, communication is preferable regardless of documentation. User requirements and desired features of the system can be converted to successful deliverables.

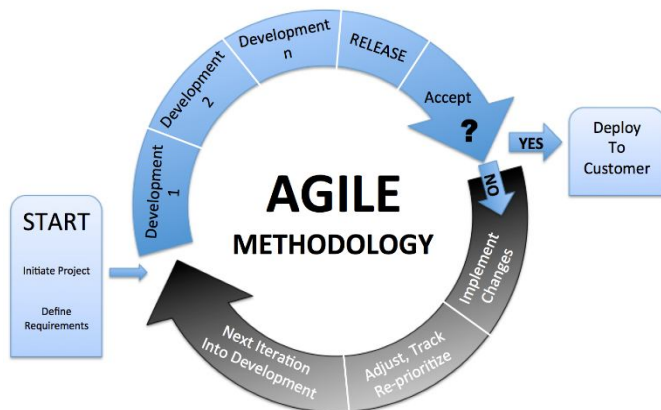
There are different frameworks under the umbrella of agile development life cycle. Like, Scrum, Feature Driven Development (FDD), Extreme Programming (XP), and Dynamic System Development Method (DSDM).

**Advantages of Agile Model**

- Promote team works and cross functional team and training
- Functionality can be developed rapidly
- Welcome changing requirements, even late in development
- People and interactions are given preferences rather than processes and tools. Developers, customers, and testers regularly interact with each other [10][11]

**Disadvantages of Agile Model**

- It does not focus on the necessary designing and documentation
- Estimating the large projects at the beginning is difficult
- Not suitable for handling complex dependencies
- Change in technology may be quite difficult and challenging task due to lack of documentation



**Figure 6: Agile Methodology**

**Basic Principle of Agile Methodology**

- Highest priority of the agile model is to satisfy the customer through the early delivery of valuable working software [8]
- The working product is the primary measure of progress
- The most efficient and effective way of communicating information is face to face communication [9]
- Cross-functional teams work together.

Table 2: A COMPREHENSIVE ANALYSIS OF SOFWTWARE PROCESS MODELS

<b>Parameters</b>	<b>Waterfall Model</b>	<b>Iterative Model</b>	<b>Spiral Model</b>	<b>Prototype Model</b>	<b>Agile Methodology</b>
<b>Requirement Specification</b>	Beginning level	Beginning level	Beginning level	Beginning level	Change incrementally
<b>Requirement Understanding</b>	Well understood	Not-Well understood	Not-Well understood	Not-Well understood	Well understood
<b>Feedback from user</b>	No	No	No	Yes	Yes
<b>Cost</b>	Low	Low	High	High	Medium
<b>Risk identification</b>	At beginning level	No	No	No	No
<b>Complexity of system</b>	No	No	Yes	Yes	No
<b>User coordination</b>	Beginning level	Medium	High	High	High
<b>Flow of activities</b>	Linear flow	Iterative flow	Identification of risk at each stage	Priority on customer feedback	Priority on customer feedback and incremental flow
<b>Precondition Or Prerequisite</b>	Complete requirements should be clearly defined	Core requirements of product should be well understood	Clear idea of quick design	No	No
<b>Usability</b>	Low	Low	Medium	High	Widely used currently
<b>Customer Priority for Implementation</b>	No	No	Intermediate	Intermediate	High
<b>Industry approach</b>	Basic	Basic	Medium	Medium	High
<b>Organization resources</b>	Yes	Yes	Yes	Yes	No
<b>Implementation time</b>	Long	Short	Depends on project	Short	Depends on project

#### 4. CONCLUSION

This study is focused on comparative analysis of different software process models. Moreover, it explained five different process models considering various factors. It's concluded that Waterfall model provides the foundation for a growth of model and the development team is familiar regarding complete requirements and environment to adopt this model. Iterative design is the enhanced model of the waterfall. Iterative design gives feedback on the previous stage. The spiral model deals with the risk of the total value of the development project and is used for the development of complex and expensive projects. The prototype model is used for when the desired system needs to have more interaction with end user. The agile methodology welcomes changes at every phase. In agile methodology, it is easy to make shifts in any feature and its focus on delivering working software frequently. The Comparative analysis includes the advantages and disadvantages of different software process models which help out to select specific model at particular situation depending on customer demand and need [10]. The feasibility of the project for development can be judged by keeping in mind the factors like cost, time and efficiency.

#### REFERENCES

- [1] SVITS, Indore MP. "A Comparative Analysis of Different types of Models in Software Development Life Cycle." *International Journal* 2.5 (2012).
- [2] Raval, Ratnmala R., and Haresh M. Rathod. "Comparative Study of Various Process Model in Software Development." *International Journal of Computer Applications* 82.18 (2013)..
- [3] Munassar, Nabil Mohammed Ali, and A. Govardhan. "A comparison between five models of software engineering." *IJCSI* 5 (2010): 95-101.
- [4] Pino, Francisco J., Flix Garcia, and Mario Piattini. "Key processes to start software process improvement in small companies." *Proceedings of the 2009 ACM symposium on Applied Computing*. ACM, 2009.
- [5] Sarker, Iqbal H., et al. "A Survey of Software Development Process Models in Software Engineering." *International Journal of Software Engineering and Its Applications* 9.11 (2015): 55-70.
- [6] Dholakia, Mr Preyash, and Mr Dishek Mankad. "The Comparative Research on Various Software Development Process Model." *International Journal of Scientific and Research Publications* 3.3 (2013).
- [7] Ashrafi, Noushin. "The impact of software process improvement on quality: in theory and practice." *Information Management* 40.7 (2003): 677-690.
- [8] Acua, Silvia Teresita, and Xavier Ferr. "Software Process Modelling." *ISAS-SCI* (1). 2001.
- [9] Kaur, Rupinder, and Jyotsna Sengupta. "Software process models and analysis on failure of software development projects." *arXiv preprint arXiv:1306.1068* (2013).
- [10] Kautz, Karlheinz, Henrik Westergaard Hansen, and Kim Thaysen. "Applying and adjusting a software process improvement model in practice: the use of the IDEAL model in a small software enterprise." *Proceedings of the 22nd international conference on Software engineering*. ACM, 2000.
- [11] Chemsripong, Sujinda, and Chutchonook Charutwinyo. "Competency Model of software Developer in Thailand." *European Journal of Molecular & Clinical Medicine* 7.3 (2020): 714-722.