IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

A Comprehensive Study of Various Software Process Models in Software Development

Dr. V. Poornima[1] and Mrs. R. Priya[2]
Assistant Professor, Department of Computer Science[1,2]
SRM Institute of Science and Technology, Chennai, India
poornimv@srmist.edu.in[1] and priyar3@srmist.edu.in[2]

Abstract: *The software development models are critical components of the entire development process. It consists of several steps, including requirement collecting, designing, development, testing, and implementation. The software process models determine the success or failure of software development. The software engineering process can be divided into two stages: At the first level, actions connected to gathering information, developing software, and maintaining it; at the second level, activity related to dentition, operation, measurement, and upgrading.This paper presents a comparative study of various software development process modelsFurthermore, it would undoubtedly serve as a directing way for the researchers to investigate new study directions. A thorough examination of all eight process models is also provided. Each software process model includes a variety of features and parameters that are simply described so that researchers can readily select a process model based on their needs, resources, and experience.*

Keywords: Software development, Process models, SDLC, Software Engineering

I. INTRODUCTION

The software development life cycle (SDLC) is a set of phases that helps everyone understand the software development process. How the software will be realised and developed, beginning with the business understanding and requirements elicitation phase, to turn these business ideas and requirements into functions and features, and continuing through its usage and operation to meet the business demands. A smart software engineer should understand how to select the SDLC model depending on the project context and business requirements.
SDLC Cycle represents the process of developing software.

Fig. 1. Life Cycle of SDLC

SDLC framework includes the following steps:
Stage 1: Planning and requirement analysis
Stage 2: Defining Requirements
Stage 3: Designing the Software
Stage 4: Developing the project
Stage 5: Testing
Stage 6: Deployment
Stage 7: Maintenance

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

To ensure project success, it may be necessary to select the appropriate SDLC model based on the project's specific problems and requirements. Furthermore, the links indicated here will help you learn more about Software Testing life cycles and SDLC phases.In this article, we will look at the many types of software process models, their benefits and drawbacks, and when to apply them.

II. SOFTWARE PROCESS MODELS

Software process model is abstract representation of theprocess which are used to develop the software, it only followsthe SDLC which includes Design, Implementation, Testing,and Maintenance.

2.1 Waterfall Model

The Waterfall Model is a sequential linear flow. In which progress is viewed as steadily descending (like a waterfall) through the phases of software implementation. This indicates that any phase of the development process can start only when the previous phase is finished. The waterfall approach does not specify how to return to a prior phase to address changes in requirements. The waterfall approach was the first and most extensively utilised approach for software development.

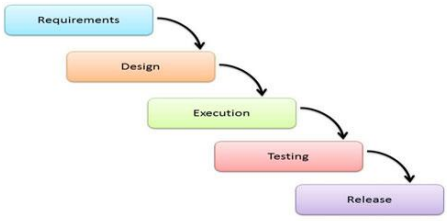


Fig. 2. Waterfall Model

A. Advantages

- Simple to explain to users and Structures method.
- Activities and stages are well specified.
- Aids in project planning and scheduling
- Early detection of errors/misunderstandings is ensured by verification at each level.
- Each phase has its own set of deliverables.

B. Disadvantages

- It is quite difficult to return to any stage after it has concluded.
- Expensive and time-consuming, in addition to the precise plan.
- Assumes that a system's requirements V-Shaped Model

2.2 V-Shaped Model

The V-model is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also known as Verification and Validation model.It is an extension of the waterfall model in which the process steps are curved upwards after the implementation and coding phases to form the usual V shape, rather than continuing down in a linear fashion. The primary distinction between the V-shaped model and the waterfall model is that the V-shaped approach includes early test planning.

A. Advantages

- It is simple and straightforward to use.
- Each Phase has its own set of deliverables.
- More likely to succeed than the waterfall model because test plans are developed early in the life cycle.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

- It works best in situations where the needs are clear.
- Product verification and validation in the early stages of development

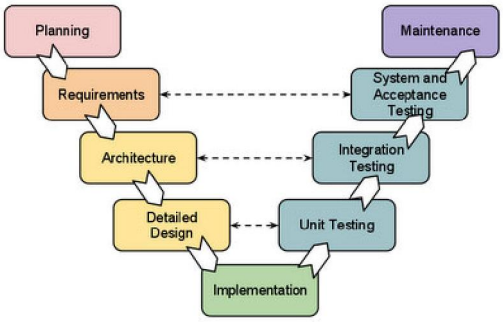


Fig.3.V-Model

B. Disadvantages

- Very rigid, such as the waterfall model
- Scope adjustment is complex and costly.
- Because the programme is built during implementation, no early prototypes are produced.
- The model lacks a clear path for problems during the testing phases.
- Expensive and time-consuming

2.3 Prototyping Model

It refers to the action of developing software application prototypes, such as incomplete versions of the software programme being developed. It is a possible activity in software development. It was used to visualise some software components in order to reduce the development team's risk of misinterpreting customer requirements.

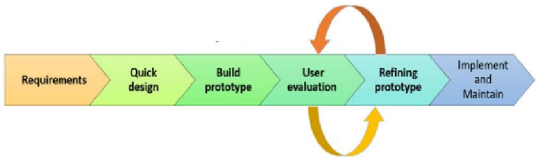


Fig.4. Prototype Model

This will also limit the number of iterations that may occur in the waterfall approach, which is difficult to apply due to its inflexibility. As a result, once the final prototype is completed, the requirement is regarded to be frozen.It has some types, such as:

- Throwaway prototyping: Prototypes that are subsequently discarded rather than becoming part of the software that is eventually deployed.

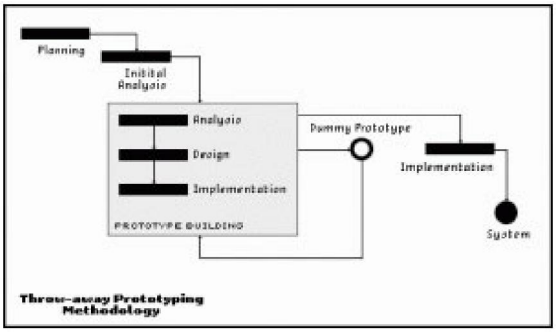


Fig. 5. Throwaway prototyping

- Evolutionary prototyping: Prototypes that progress into the final system through iterative user feedback inclusion.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

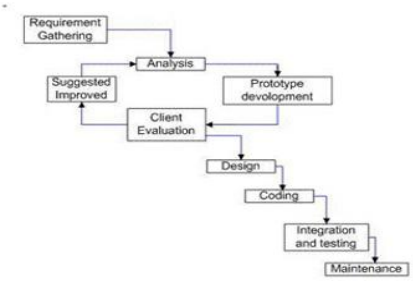


Fig. 6. Evolutionary prototyping

- Incremental prototyping: Separate prototypes are used to create the final product. Finally, the individual prototypes are combined to form an overall design.

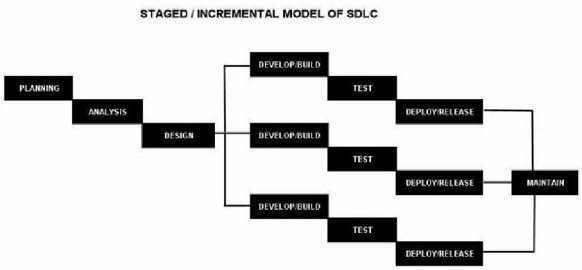


Fig.7. Incremental prototyping

- Extreme prototyping: Primarily utilised in online applications. It basically divides web development into three phases, each one building on the previous one. The initial phase is a static prototype made up mostly of HTML pages. The screens are programmed and completely functioning in the second phase utilising a simulated services layer. The services are implemented in the third phase.

A. Advantages

- Saves time and money, although this can backfire if the developer wastes time constructing prototypes.
- Improved and improved user participation

B. Disadvantages

- Inadequate analysis. The prototype final system caused user confusion.
- Misunderstanding of user objectives by the developer
- Excessive prototype development time
- The prototypes are expensive to implement.

2.4 Spiral Model

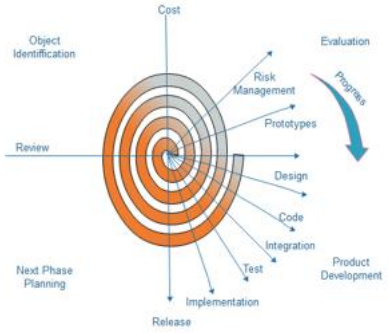
The spiral model is an evolutionary software process model that combines prototyping's iterative feature with the linear sequential model's controlled and systematic aspects. It implements the potential for rapid development of new software versions. The programme is developed in a series of incremental releases using the spiral methodology. The supplementary release may be a paper model or prototype during the early phases. Later iterations result in increasingly full versions of the engineered system.

Each spiral cycle is divided into four parts:

- Setting objectives: Each cycle in the spiral begins with the identification of the cycle's purpose, the many choices for accomplishing the targets, and the constraints that exist.
- Risk assessment and reduction: The cycle's next step is to calculate these numerous options depending on the goals and restrictions. The focus of evaluation at this stage is on the project's risk perception.
- Development and validation: The following stage is to create methods to address uncertainties and hazards. Benchmarking, simulation, and prototyping are examples of activities that may be included in this process.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

- Finally, the following move has been planned. The project is reviewed, and the decision is made whether to proceed with it.



- Fig. 8. Spiral Model

A. Advantages

- Estimates (e.g., budget, schedule, etc.) become more realistic as work progresses because critical issues are identified earlier.
- Early developer engagement
- Manages risks and phases the system's development

B. Disadvantages

- Expensive and lengthy production process.
- Requires specialised knowledge to assess risks and assumptions.
- Highly personalised, restricting reusability

2.5 Incremental Model

The incremental model is a software development process in which requirements are separated into many isolated modules of the software development cycle. Each module in this paradigm goes through the processes of requirements, design, implementation, and testing. Every succeeding module release adds functionality to the preceding iteration. The technique is repeated until the entire system is completed.

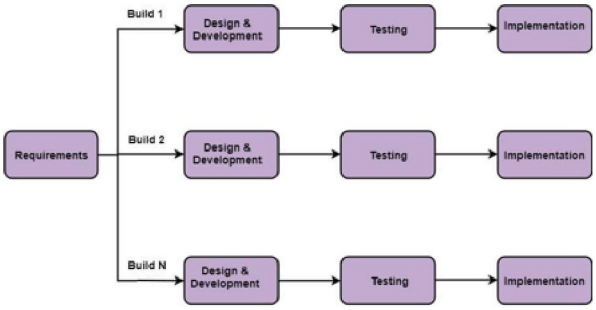


Fig. 9. Incremental model

The incremental model's many phases are as follows:

1. Requirement analysis: The product analysis expertise identifies the requirements in the first phase of the incremental model. The requirement analysis team also understands the system functional requirements. This phase is critical in the incremental paradigm of software development.
2. Design & Development: The design of the system functionality and the development technique are completed successfully in this phase of the SDLC Incremental model. When new functionality is added to software, the incremental model employs style and development phase.
3. Testing: The testing phase in the incremental model evaluates the performance of each current function as well

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

as added capabilities. Various methods are utilised to test the product during the testing process.

4. Implementation: The implementation step supports the development system's coding phase. It includes the final coding that is designed during the designing and developing phase and tested during the testing phase. Following the completion of this phase, the number of working products is increased and updated all the way to the final system product.

A. Advantage

- Errors are easily identified.
- Easier to test and troubleshoot
- More adaptable
- Risk management is simple because it is handled during iteration.
- The Client receives critical functionality early on.

B. Disadvantage

- Requires careful planning
- The overall cost is high.
- Module interfaces must be well defined.

2.6 Rapid Application Development

The Rapid Application Development (RAD) paradigm relies on constant user interaction in the requirement gathering process via prototyping. If the requirements are clearly understood and articulated, and the project scope is small, the RAD technique enables a development team to quickly build a fully functional system.

Rapid Application Development (RAD) is the concept that products can be generated more quickly and with higher quality by:

- Collecting requirements via workshops or focus groups
- Design prototyping and early, iterative user testing
- Reusing software components
- A tight schedule for referring design enhancements to the next product release.
- Team meetings and reviews are less formal

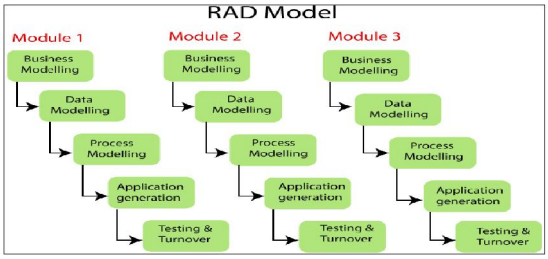


Fig.10. RAD model

The following are the stages of RAD:

1. 1.Business Modeling: The information flow between business operations is defined by addressing questions such as what data drives the business process, what data is generated, who generates it, where the information goes, who processes it, and so on.
2. Data Modeling: The data gathered through business modelling is refined into a set of data objects (entities) required to serve the business. The attributes (characteristics of each entity) are defined, as is the relationship between these data items (entities).
3. Process Modeling: The data objects defined during the data modelling phase are changed to achieve the data flow required to operate a business function. Processing descriptions are written for the purposes of adding, updating, removing, or retrieving data.
4. Application Generation: Automated tools are used to aid in the development of software; they, too, employ 4th GL approaches.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

5. Testing and Turnover: Because RAD emphasises reuse, many of the programming components have already been tested. This cuts down on overall testing time. However, the new part must be thoroughly tested, as must all interfaces.

A. Advantages

- This model is adaptable to new circumstances.
- Changes are possible in this model.
- Each phase of RAD provides the customer with the highest priority functionality.
- It shortened development time.
- It promotes feature reusability.

B. Disadvantages

- It necessitated the use of extremely skilled designers.
- Not all applications are RAD compatible.
- We cannot employ the RAD paradigm for smaller projects.
- It is not suited due to the considerable technical danger.
- User participation is required.

2.7 Iterative Model

This Model begins with certain software specs and progresses to the creation of the initial version of the software. If there is a need to alter the programme after the first version, a new version of the software is developed with a new iteration. Every Iterative Model release is completed in an exact and definite period known as iteration. The Iterative Model enables access to prior phases in which variations were made. The project's final product at the end of the Software Development Life Cycle (SDLC) process

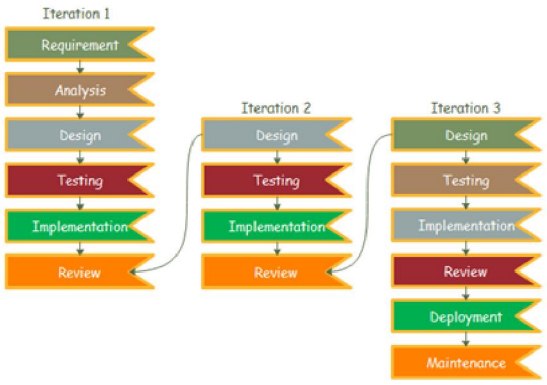


Fig. 11.Iterative model

The various phases of Iterative model are as follows:

1. Requirement gathering &analysis:In this phase, requirements are gathered from customers and checked by an analyst to see if they will be met. Analyst determines whether or not the need will be met within the budget. Following this, the software team moves on to the next phase.
2. Design: During the design phase, the team creates the software using various diagrams such as a data flow diagram, an activity diagram, a class diagram, a state transition diagram, and so on. **Tech Cheap**
3. Implementation: During implementation, requirements are defined in a coding language and converted into computer programmes known as software.
4. Testing: Following the completion of the development process, software testing begins utilising various test methods. The most frequent test methods are white box, black box, and grey box.
5. Deployment: Following the completion of all processes, software is deployed to its working environment..

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

6. Review: Following product deployment, the review phase is undertaken to examine the behaviour and correctness of the generated product. If an error is discovered, the process is restarted at the requirement collection stage.
7. Maintenance: During the maintenance phase, after the programme has been deployed in the working environment, there may be some issues, errors, or new upgrades that are necessary. Debugging and new addition choices are part of maintenance.

A. Advantages

- A parallel development is possible.
- It is easily adaptable to the project's ever-changing requirements.
- During iteration, risks are discovered and resolved.
- Documentation time is limited, while design time is not.

B. Disadvantages

- It is not appropriate for minor tasks.
- Additional resources may be necessary.
- Because of incomplete criteria, the design can be revised multiple times.
- Changes in requirements might lead to budget overruns.
- Because to changing needs, the project completion date has not been confirmed.

2.8 Agile Model

It is built on iterative and incremental development, with requirements and solutions evolving through cross-functional team engagement.It may be utilised for any type of project, but it requires more customer engagement and interaction. It can also be utilised when the customer requires a functional need completed in less than three weeks and the criteria are unclear.

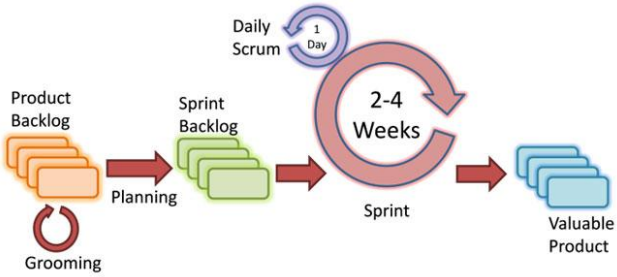


Fig. 12.Agile model

Following are the phases in the Agile model:

1. Gathering requirements: During this step, you must define the criteria. You should describe business prospects and estimate the time and effort required to complete the project. You can assess the technical and economic viability based on this information.
2. Design the needs: Once the project has been selected, collaborate with stakeholders to create the requirements. You can use a user flow diagram or a high-level UML diagram to demonstrate the functionality of new features and how they will interact with your existing system.
3. Construction/iteration: Work begins when the team determines the requirements. Designers and developers begin work on their project, which seeks to deliver a functional product. The product will go through several rounds of development, thus it will have simplistic, minimal functionality.
4. Testing: The Quality Assurance team checks the product's performance and looks for bugs during this step.
5. Deployment: During this phase, the team releases a product for the user's workplace.
6. Feedback: The final stage after releasing the product is feedback. In this stage, the team receives product feedback and works through it.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

Following are the methods in the Agile model:

Agile Testing Methods:

- Scrum: SCRUM is an agile development process that focuses on task management in team-based development environments.
- eXtremeProgramming(XP): This practise is utilised when customers' expectations or requirements are continually changing, or when they are unsure about the system's performance.
- Crystal:This approach is comprised of three concepts:
  1. Chartering: This phase includes a variety of activities such as assembling a development team, conducting feasibility study, developing plans, and so on.
  2. Cyclic delivery: this includes two more cycles, which are as follows:
  A. The team revises the release schedule.
  B. The integrated product provides services to users.
  3. Wrap up: This step performs deployment and post-deployment depending on the user situation.
- Dynamic Software Development Method(DSDM): DSDM is a rapid application development technique that provides an agile project distribution framework for software development. The key aspects of DSDM are that users must be actively linked, and teams have been granted decision-making authority. DSDM employs the following techniques:
  1. Time Boxing
  2. MoSCoW Rules
  3. Prototyping
- Feature Driven Development(FDD):
  1. The "Designing and Building" features are the centre of this method. Unlike other smart methods, FDD describes the small steps of work that must be obtained separately for each function.
  2. Lean Software Development: The lean software development process adheres to the "just in time manufacturing" premise. The lean method denotes boosting the speed of software development while decreasing costs. Lean development can be divided into seven stages.

A. Advantages

- Frequent Delivery
- Face-to-Face Communication with clients.
- Efficient design and fulfils the business requirement.
- Anytime changes are acceptable.
- It reduces total development time

B. Disadvantages

- Shorten the time it takes to access certain system capabilities.
- There is no room for speculation with face-to-face communication and continual feedback from the customer representative.
- The final result is high-quality software delivered in the shortest amount of time and a delighted customer.

III. CONCLUSION

This research focuses on a comprehensive study of various software process models. This paper's contribution is a detailed survey of eight commonly used process models. First one is Waterfall model whichprovides base for other development models. Then its enhanced models are explained inV shaped model, Incremental Model, Prototyping Model, Spiral model, RAD model, Iterative modeland finally, Agile development model.Furthermore, this research includes the advantages and disadvantages of different models which canhelp to select specific model at specific situation depending on customer demand.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, August 2022
Impact Factor: 6.252

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]. M. Völter, et al., "Model-driven software development: technology, engineering, management", John Wiley & Sons, (2013).

[3]. Chemsripong, Sujinda, and ChutchonookCharutwinyo."Competency Model of software Developer inThailand." European Journal of Molecular & ClinicalMedicine 7.3 (2020): 714-722.

[4]. SVITS, Indore MP. "A Comparative Analysis ofDifferent types of Models in Software Development LifeCycle." International Journal 2.5 (2012).

[5]. Raval, Ratnmala R., and Haresh M. Rathod."Comparative Study of Various Process Model inSoftware Development." International Journal ofComputer Applications 82.18 (2013).

[6]. Chemsripong, Sujinda, and ChutchonookCharutwinyo"Competency Model of software Developer inThailand." European Journal of Molecular & Clinical Medicine 7.3 (2020): 714-722.

[7]. Sajid Ahmed Ghanghro1, Dr. Muhammad Ajmal Sawand,Wajid Ahmed Channa, Ubaidulla aliasKashif, Muhammad Hanif Tunio, Kishor Kumar, Nooruddin,"Comparative Analysis of Software Process Models inSoftware Development",ISSN 2278-3091,Volume 10, No.3, May - June 2021 International Journal of Advanced Trends in Computer Science and Engineering

[8]. https://www.javatpoint.com/software-processes