RESEARCH ARTICLE OPEN ACCESS

Models in Software Engineering – An Introduction

Avtar Singh, Suman Lata, Harpreet Kaur
Guru Kashi University Talwandi Sabo
@hmann5917@gmail.com, @suman_cse@ccet.ac.in

Abstract

The idea of models is essential to software engineering. The term is defined and examined in this essay from a number of angles. Examples of the most significant model categories are shown. Models are extremely helpful, but they may also be risky, especially when used carelessly. These issues are displayed. Finally, the significance of models in research on software engineering is highlighted.

Keywords – Waterfall Model , V-Model, Incremental Model, RAD Model, Agile Model, Spiral Model, Prototype Model

Software Engineering

A field of engineering known as "software engineering" focuses on the development of software products using established scientific concepts, methods, and practices. An efficient and dependable software product is the end result of software engineering.

Types of Software Development Models:

1. Waterfall Model

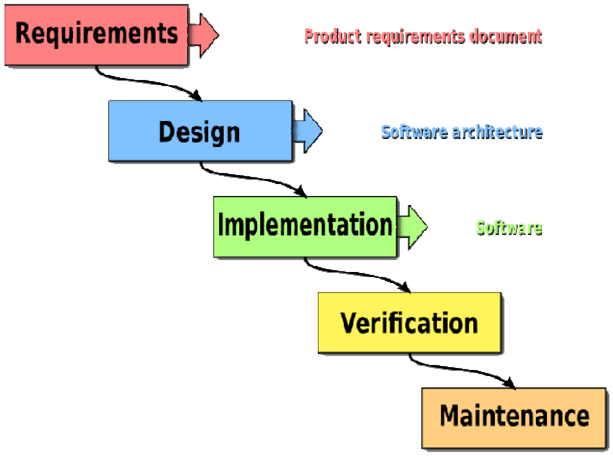

Figure 1 Waterfall Model

The Waterfall Model is a linear sequential flow that's frequently applied to projects with clear criteria. The model's process moves gradually downhill through the stages of software implementation, reflecting its name.

This methodology requires that the prior phase be finished before starting the subsequent. The Waterfall Model is best suited for projects that do not anticipate unanticipated changes mid-development because it does not enable reverting back to previously finished phases.

Advantages and Disadvantages of Waterfall

Advantage: The Waterfall Model is a controllable approach that is excellent for the lifecycle management of smaller projects when the needs are understood and decided upon up front. It is clear and simple.

Disadvantages: The Waterfall Model's strict structure makes it unsuitable for complicated projects when there is a possibility of requirement changes and/or extensive impromptu testing throughout the software development phase.[1]

2. V-Model

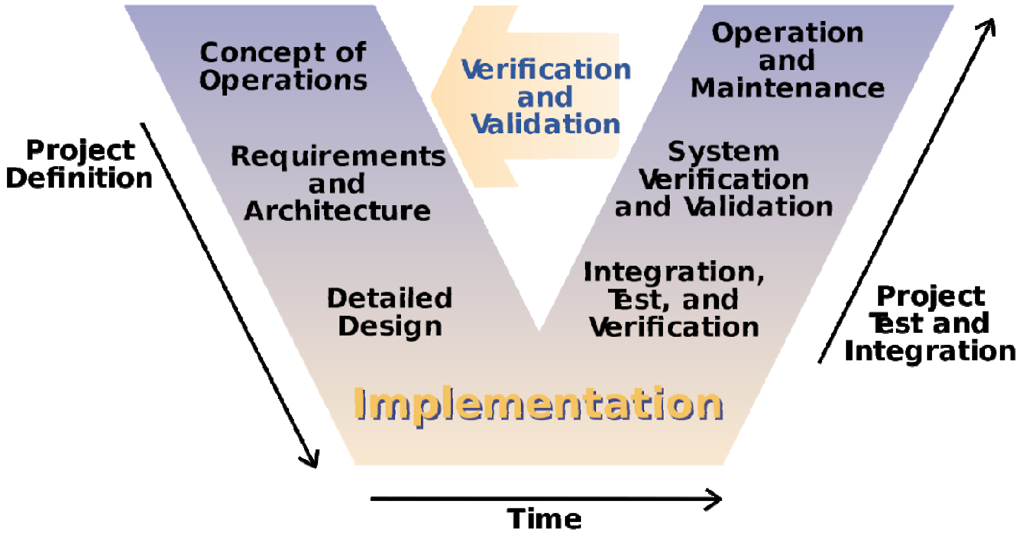
Concept of Operations
Verification and Validation
Operation and Maintenance
Project Definition
Requirements and Architecture
System Verification and Validation
Detailed Design
Integration, Test, and Verification
Project Test and Integration
Implementation
Time

Figure 2 V Model

The V-Model, which is a development of the waterfall model, has sequential flow capabilities as well. However, following the coding, the software development lifecycle bends upward rather than continuing to move exclusively linearly downward.

There is a testing phase in the subsequent ascending sequence for each step in the downward process. When software needs and tool requirements are known in advance, this paradigm is employed.

Advantages and Disadvantages of V-Model

Advantages:The V-Model is a straightforward method that works well for minor tasks. The development stage's test plans and the V-regularly Model's lifecycle's scheduled upgrades might increase the likelihood of success.

Disadvantages: The V-Model, which is extremely rigid by nature like the Waterfall Model, isn't suited for applications or systems software that may need unanticipated changes or upgrades during the software lifespan.[2]

3. Incremental Model

The Incremental Model uses iterative and incremental development stages to address the drawbacks of the Waterfall Model. In essence, the incremental model is made up of multiple little waterfall cycles.

Advantages and Disadvantages of Incremental Model

Advantages:For projects that require certain modification requests to be accommodated in between increments, the incremental model is a fantastic option. The ability to identify issues sooner in the software development process allows for better lifecycle management planning, which is another advantage of this paradigm.

Disadvantages: The Incremental Model's possible drawback is the requirement for strategic planning and documentation. Additionally, this approach typically calls for additional personnel and financial support for the project. This paradigm isn't the best for continuous development because the subsequent sequence can't start until the preceding one is finished in full.

4. RAD Model

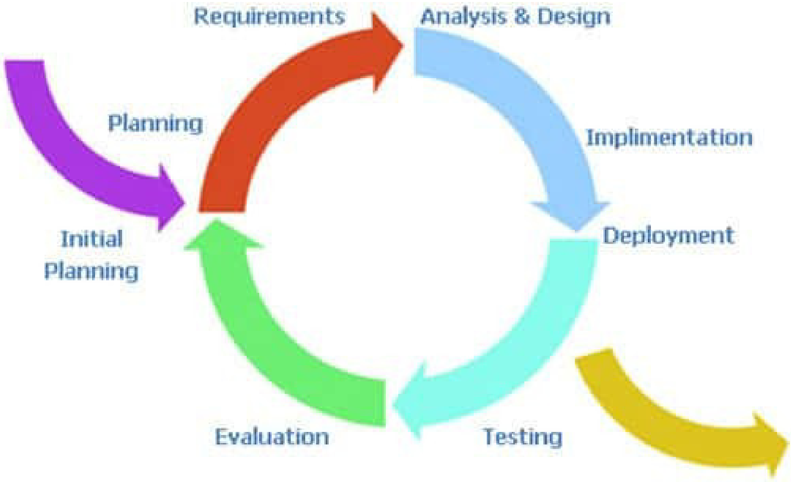

Figure 3 RAD Model

The RAD Model, often known as rapid application development, is an alteration of the incremental model. When using this methodology, a number of components are created concurrently as though they were separate, smaller projects. The various parts are then put together to create functional prototypes.

Advantages and Disadvantages of RAD

Advantages: The RAD Model provides for quicker software development and more customer input at every stage of the process.

Disadvantages:The RAD Model's applicability is constrained since the project must be easily modularized into numerous stages. Additionally, it needs developers with extensive experience as well as great modelling and planning abilities. Problems with the final assembly of the components might lead to unanticipated setbacks and the need to redesign certain parts to better suit the others.

5. Agile Model

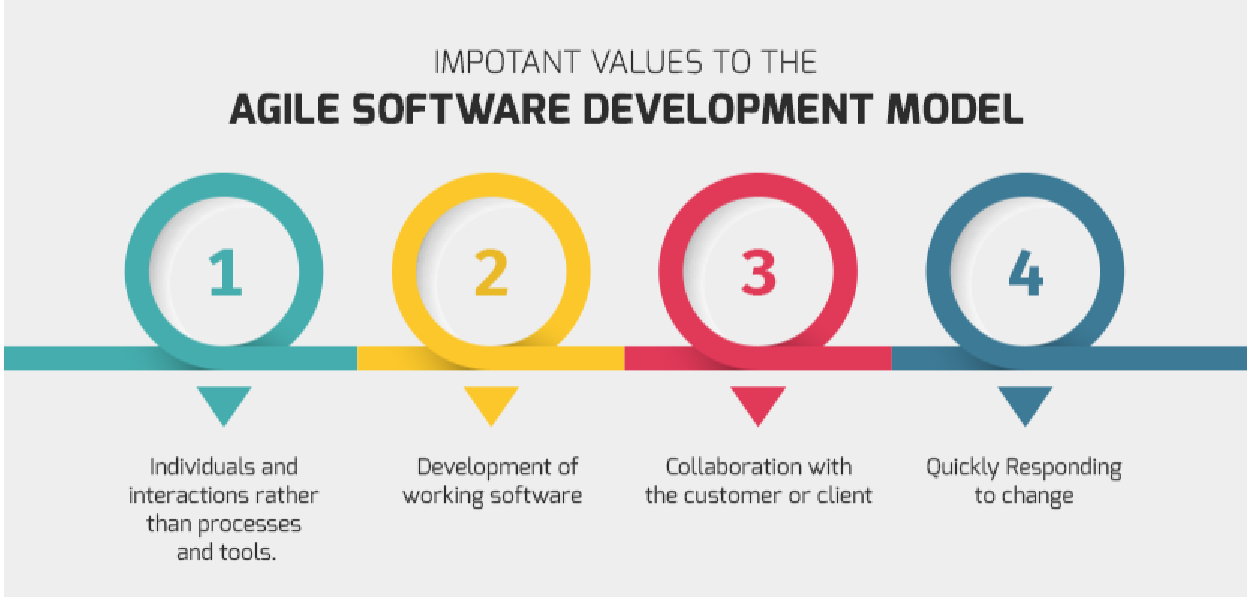
IMPOTANT VALUES TO THE
AGILE SOFTWARE DEVELOPMENT MODEL
1
2
3
4
Individuals and interactions rather than processes and tools.
Development of working software
Collaboration with the customer or client
Quickly Responding to change

Figure 4 Agile Model

This paradigm is founded on user interaction, process flexibility, and the quick delivery of functional software components. It breaks down product development into tiny incremental builds that are subsequently supplied in iterations, therefore it includes both incremental and iterative capabilities.

Advantages and Disadvantages of Agile

Advantages:The time it takes to produce a single system feature is slashed thanks to the agile model. It also necessitates extensive communication and ongoing user feedback that may give the project a clear direction.

Disadvantages: As it depends on end-user involvement, which may or may not be clearly articulated, the Agile approach has the potential to deviate from its intended course. For an Agile software development method, documentation is also limited and a knowledgeable, cross-functional team is needed.

6. Iterative Model

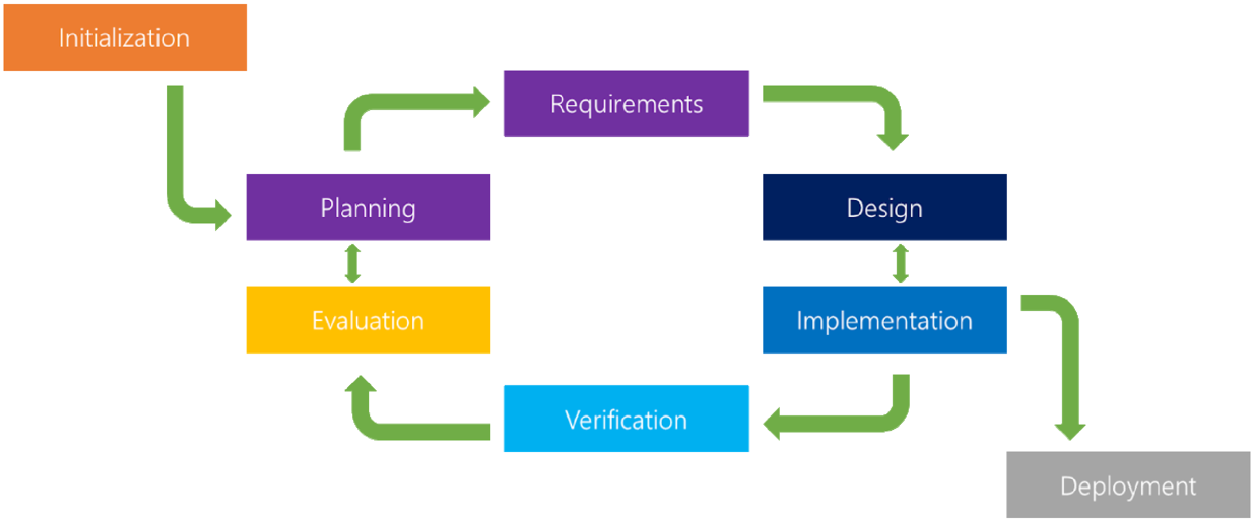

Figure 5Iterative Model

Instead of attempting to start with a thorough specification of requirements, the iterative model concentrates on describing and creating particular software components. In an iteration, a preliminary product is produced, reviewed, and refined in the following iteration, and so on. The iterative model depends on the gradual development of the entire product.

Advantages and Disadvantages of Iterative Model

Advantages:When employing this software development strategy, it is simple to spot issues early on since the product is created gradually.

Disadvantages:The Iterative Model can take longer and be more expensive because each iteration phase is rigid and has no overlaps.[3]

7. Spiral Model

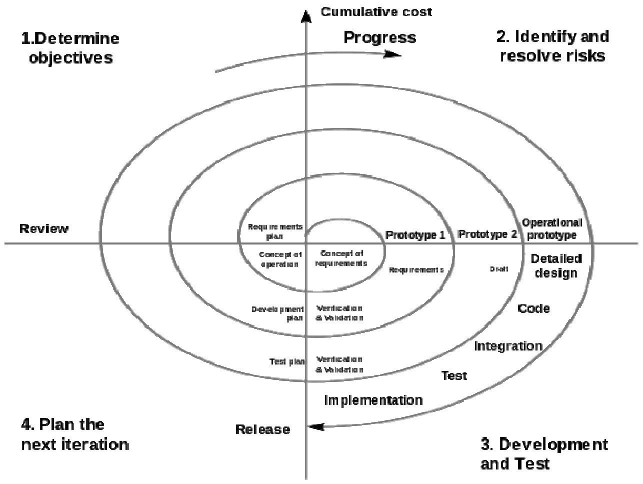

Figure 6 Spiral Model

In an effort to combine the benefits of top-down and bottom-up production, the Spiral Model incorporates components of both the Iterative and Waterfall development methods. Identification, Design, Construct/Build, Evaluation, and Risk Analysis are the four phases of the Spiral Model. In spiral-style iterations, the software project constantly moves through these stages.

Advantages and Disadvantages of Spiral Model

Advantages: The Spiral Model can be useful since it separates development into phases and controls risks. Due to earlier discovery of obstacles, projections for budget and schedule are also more accurate.

Disadvantages: Because of how heavily specialised this approach is, reusing the process might be challenging. It also needs team members who are skilled in risk assessment.

8. Prototype Model

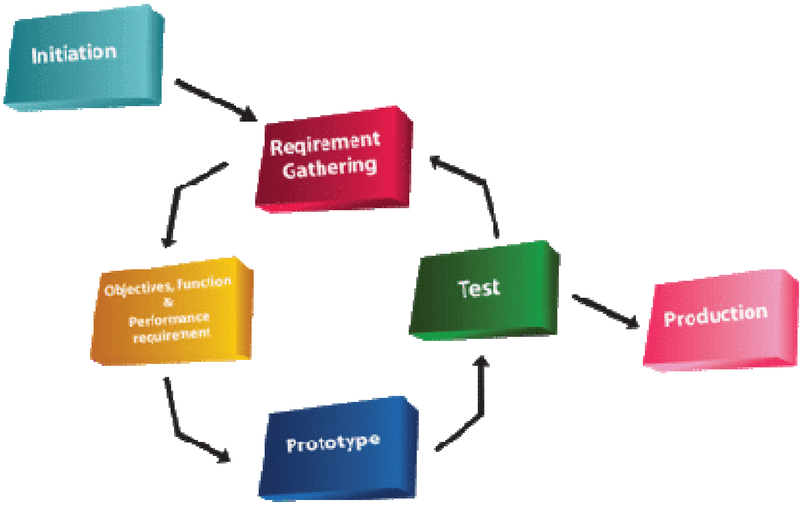

Figure 7 Prototype Model

Because of how heavily specialized this approach is, reusing the process might be challenging. It also needs team members who are skilled in risk assessment. The Prototype Model depends on building prototypes of the system or application software that is used to visualise different software components. By allowing for significant user engagement and input, this paradigm helps to reduce the gap between misunderstandings of needs.

Advantages and Disadvantages of Prototype Model

Advantages:The Prototype Model can result in decreased costs and time with significant user participation.

Disadvantages:The user may become confused by this paradigm between the prototype and the completed product.

Parameter	Waterfall Model	Incremental Model	Prototype Model	RAD Model	Spiral Model	Agile Model	V Model
Clear Requirement Specifications	Initial level	Initial level	At medium level	Initial level	Initial level	Change incrementally	Initial level
Feedback from user	No	No	Yes	No	No	No	Yes
Speed to change	Low	High	Medium	No	High	High	High
Predictability	Low	Low	High	Low	Medium	High	High
Risk identification	At initial level	No	No	No	Yes	Yes	Yes
Practically implementation	No	Low	Medium	No	Medium	High	High
Any variation done	Yes V model	No	No	No	Yes win spiral	No	No
Understandability	Simple	Intermediate	Intermediate	Intermediate	Hard	Much Complex	Intermediate
Resource organization	Yes	Yes	Yes	Yes	No	No	Yes
Usability	Basic	Medium	High	Medium	Medium	Most use now a days	Medium
Customer priority	Nil	Nil	Intermediate	Nil	Intermediate	High	Intermediate
Industry approach	Basic	Basic	Medium	Medium	Medium	High	Medium
Cost	Low	Low	High	Very high	Expensive	Much Expensive	High
Elasticity	No	No	Yes	Yes	No	Very high	Medium

Reference:

[1]. https://ca.insight.com/en_CA/content-and-resources/2016/07152016-types-of-software-development-models.html

[2]. https://flexagon.com/blog/7-software-development-models-you-should-know/

[3]. https://ca.insight.com/en_CA/content-and-resources/2016/07152016-types-of-software-development-models.html

[4]. Lata, S., and R. Kumar. “A Hybrid Approach for ECG Signal Analysis.” *Proceedings - IEEE 2018 International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2018*, 2018, doi:10.1109/ICACCCN.2018.8748858.

[5]. Lata, Suman, and Rakesh Kumar. “Disease Classification Using ECG Signals Based on R-Peak Analysis with ABC and ANN.” *International Journal of Electronics, Communications, and Measurement Engineering*, vol. 8, no. 2, July 2019, pp. 67–86, doi:10.4018/IJECME.2019070105.

[6]. Lata, Suman, and Dheerendra Singh. "A Hybrid Approach for Cloud Load Balancing." In 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), pp. 548-552. IEEE, 2022.