MODELS AND METHODOLOGIES SOFTWARE DEVELOPMENT

Abdulakhatov Muzaffarjon

Associate professor of the "Industrial Management and Digital Technology" department of the International Nordic University
muzaffar.abdulakhatov@gmail.com

Abstract

Software development is a complex process that requires structured approaches to ensure the creation of high-quality and reliable software. Models and methodologies serve as frameworks to guide developers through the stages of software development, from initial requirements gathering to final deployment and maintenance. This abstract explores the key models and methodologies utilized in contemporary software development, highlighting their purposes, characteristics, and use cases.

Keywords: Software Development, **Cascade model**, V-shaped model, Spiral Model, DevOps, Model-Driven Development

Introduction

The life process of any system or software product can be described using a life cycle model consisting of stages. Models can be used to represent the entire life cycle from conception to retirement, or to represent a portion of the life cycle relevant to the current project. The life cycle model is represented as a sequence of stages that can overlap and/or repeat cyclically according to application, size, complexity, need for change, and capabilities. Each stage described wording goals and exits. Processes and life cycle activities are selected and executed at these stages to fully satisfy the purpose and results of each stage. Different organizations may use different stages within the life cycle. However, each stage is implemented by the organization responsible for that stage, with due consideration of the information available in life cycle plans and decisions made in previous stages. Similarly, the organization responsible for the current stage maintains records of decisions made and records of assumptions related to subsequent stages of that life cycle.

The software life cycle model is understood as a structure that defines the sequence of execution and relationships between processes, actions and tasks on throughout the life cycle. The life cycle model depends from specification, scale and difficulties project and specifications conditions, in which system is created and functioning. Model J C BY includes: stages, results of work at each stage, key events - points of completion of work and decision making. The life cycle model of any specific software determines the nature of the process of its creation, which is a set of works ordered in time, interconnected and combined into stages, the implementation of which is necessary and sufficient to create software that meets the specified requirements. A stage is understood as part of the creation process BY, limited certain temporary within ending with the release of a specific product (models, software components, documentation), determined by the requirements specified for this stage.

Cascade model

The first model to become widely known and truly structure the development process is the cascade (waterfall) model. Each stage of the waterfall model ends with some results that serve as input for the next stage. The requirements for the software being developed, determined at the stage of requirements formation, are strictly documented in the form of technical specifications and are recorded for the entire duration of the project development.

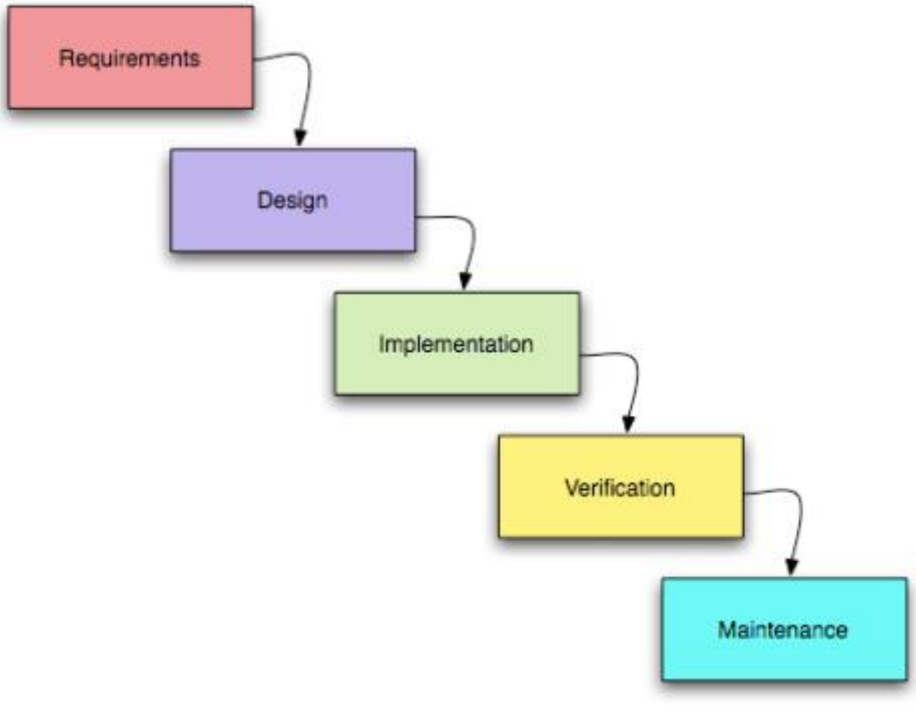

Figure. 1. Standard waterfall model

Cascade model Maybe be used at creation BY, for which V himself began development Can enough exactly and fully formulate All requirements. IN That same time this an approach has a number of disadvantages, caused primarily by the fact that the actual creation process BY never fully Not was packing V such tough diagram.

A short time after its birth, the cascade model was modified by Winst Royce, taking into account the interdependence of the stages and the need to return to previous stages, What Maybe be caused by for example, incompleteness requirements or errors in the formation of the task. The process of creating software is usually iterative: the results of the next stage often cause changes in design solutions developed at earlier stages. Thus, there is a constant need to return to previous stages and clarify or revise previously made decisions. As a result, the real software creation process takes the form shown in Figure . *1* .

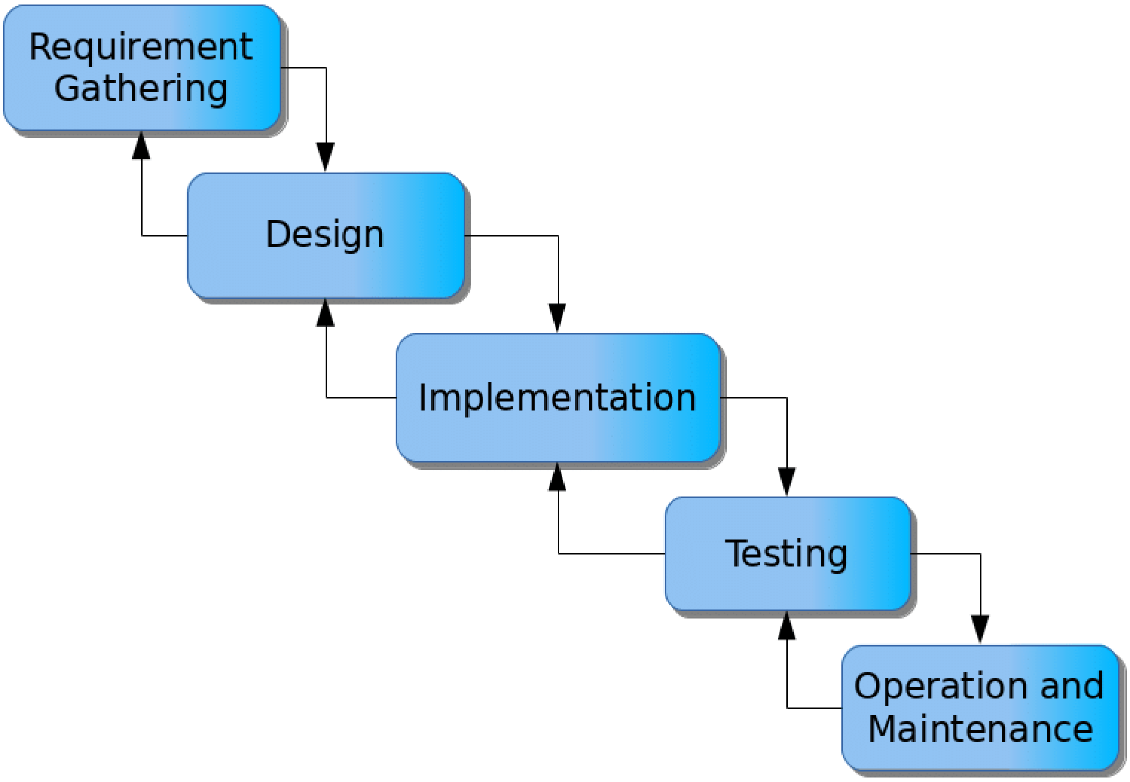

Figure 2. Modified waterfall model

The most common result of a waterfall approach to software development is late failure. Projects seem to be progressing Fine, but only before those since then Bye Work Not will enter in the final stage, And Then it turns out What consumers dissatisfied with the created product.

V-shaped model, How variety cascade model

The basic principle of the V-shape model is that the detail of a project increases from left to right along with the passage of time, and neither can be reversed. Iterations in the design are made horizontally, between the left and right sides of the letter.

V-model is a variation of the waterfall model , in which development tasks go from top to bottom along the left side of the letter V, and testing tasks go up along the right side of the letter V. Horizontal lines are drawn inside the V lines, showing How results each from development stages influence the development of the testing system at each testing stage. The model is based on the fact that acceptance testing is based primarily on requirements, system testing – on requirements And architecture, comprehensive testing - on requirements, architecture and interfaces, and component testing – on requirements, architecture, interfaces And algorithms.

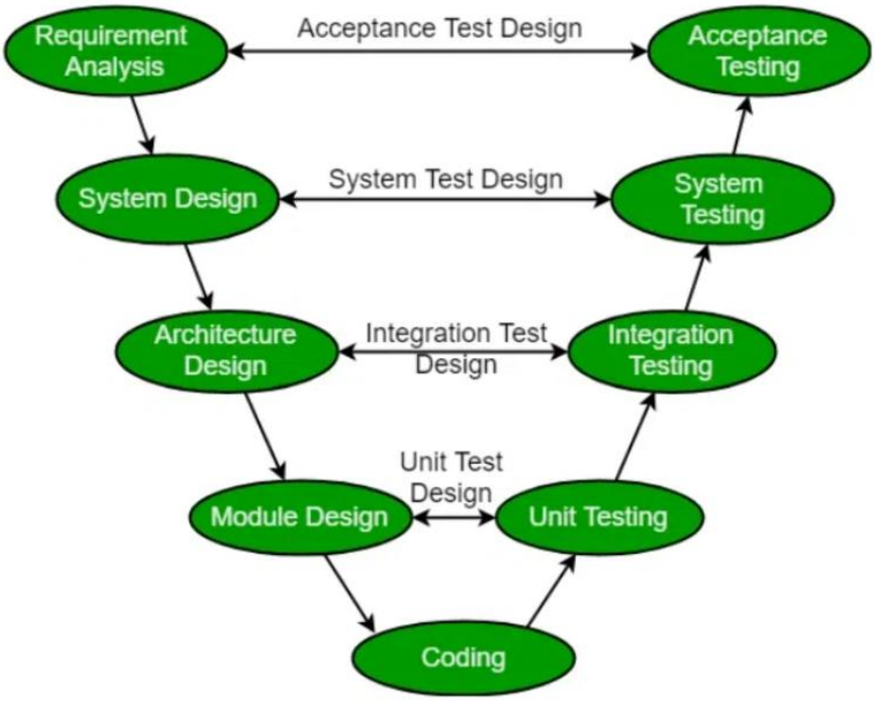

Figure 3. V-shaped model

A feature of this model is the division of stages into three logical stages: design (detailing requirements), implementation, testing. The V-Model provides guidance to organizations and project teams for executing and completing projects in a consistent and repeatable manner. Application of the principles of the V-model ensures that user requirements are identified and captured. The approved requirements can be translated into functions of the finished application, (And) application reflects requirements users.

Iterative model

The iterative model involves dividing the project life cycle on subsequence iterations, each from which reminds

"mini-project", including all phases of the life cycle as applied to the creation of smaller pieces of functionality compared to the project as a whole. The goal of each iteration is to obtain a working version of the software system, including functionality defined by the integrated content of all previous and current iterations. The result of the final iteration contains all the required functionality of the product. Thus, with the completion of each iteration, the product develops incrementally.

The chances of successfully creating a complex system will be greatest if it is implemented in a series of small steps and if each step contains a clearly defined outcome, as well as the ability to return to the results of a previous successful iteration if it fails. Before putting into action all the resources intended for creating software, the developer has the opportunity to receive feedback from real world (customers, users) and correct possible errors in the project.

Incremental model

The idea behind the incremental model is that software system should develop by principle increments, so that the developer can use data obtained during the development of earlier versions of the software. New data is obtained both during software development and during its use, where possible. The key steps in this process are simply implementing a subset of the software requirements and refining the model over a series of successive releases until the software is fully implemented. During each iteration, the organization of the model changes and new functionality is added to it.

To organize incremental development, a characteristic time interval is usually selected, for example, a week. Then, during this interval, the project is updated: new documentation, both text and graphic, is added, the set of tests is expanded, are added new software codes and T. d. In theory Development steps can be carried out in parallel, but such a process is very difficult to coordinate. Incremental development works best if the next iteration begins after all artifacts V previous iterations finished, and significantly It's worse if the time required to update artifacts significantly exceeds the selected interval.

Conclusion

Selecting the appropriate model or methodology depends on various factors, including project size, complexity, team dynamics, and the nature of requirements. While traditional models like Waterfall and V-Model offer structured approaches for projects with stable requirements, Agile methodologies and DevOps provide the flexibility needed for modern, dynamic software development. The ongoing evolution of these methodologies reflects the continuous quest for more efficient, adaptive, and effective software development practices.

References.

1. Decree of the President of the Republic of Uzbekistan dated February 28, 2023 No. PF-27 "New Development Strategy of Uzbekistan for 2022-2026". Tashkent city, 2023.
2. Abdulakhatov M. M, “The main impacts of digital transformation on small to medium-sized business enterprises” Scientific electronic journal of Digital Economy and Information Technologies No. 2, 2022
3. Abdulakhatov M. M “The digital economy is the basis for economic development” Scientific journal of Economy and Society No. 10,October 7, 2022
4. Kobilov A.U, Abdulakhatov M.M, Rajabov Sh.B, “Artificial intelligence as a technological innovation for economic development of the republic of Uzbekistan” ICFNDS ’22, December 15, 2022, Tashkent, TAS, Uzbekistan.
5. Kevin, Roebuck. System Development Life Cycle, Washington.2007
6. Tata McGraw. Software Engineering, Washington.2007