FACULTY OF COMPUTING

SEMESTER 1 2024/2025

SECP3204 SOFTWARE ENGINEERING

SECTION 01

ASSIGNMENT 1: WATERFALL METHODOLOGY

LECTURER: DR. ARYATI BINTI BAKRI

Student Name	Matric No.
NURUL ADRIANA BINTI KAMAL JEFRI	A23CS0258
TEH RU QIAN	A23CS0191

Waterfall Methodology

Nurul Adriana binti Kamal Jefri, Teh Ru Qian

The following essay addresses the Waterfall model, which is a traditional software development process that involves conducting analyses, designs, and implementations followed by testing and maintenance in a straight order. To maintain systematic advancement, the cycle does not allow for proceeding to the next step without the successful completion of the preceding one. In addition, through case studies and literature reviews, the research presents a detailed analysis of each section of the model and highlights the applicability of the model for projects with fixed requirements and its limitations for projects with changing or complex systems. The findings praise important attributes of the model, such as the structure which ensures good documentation of the project and the provision of timelines, but also indicate important limitations, for example, rigidity in accommodating changes during the project. After conducting a comprehensive analysis, we examine and elucidate the most suitable circumstances under which the Waterfall technique can be utilized and propose alterations to it for application in contemporary project situations.

***Keywords*—** The waterfall methodology, the life cycle of software development, development in sequence, management of projects, model of linear processes, analysis of requirements, and structured documents.

I. Introduction

The Waterfall approach is one of the oldest methods of software Building and it has been in existence since the 1970s as a way of organizing and systematizing complicated project processes. Waterfall is generally appropriate for work where the requirements do not change and the scope has been fixed; owing to it containing a straightforward linear progression of defined phases that consist of analysis, design, implementation, testing and finally maintenance.

There are little common areas of the different steps, in such a way that each step becomes the foundation of the next one in such a way that every recording and evaluation is done thoroughly before proceeding. Even though its inflexibility may prove hard when the requirements change, the regulatory control and stability that this model provides is highly appealing in the sectors that have stringent regulations in place.

This research evaluates the limits of the Waterfall framework against very many project types, pointing out both positive and negative aspects of its use in the current software development processes, which is centered on speed.

The Waterfall technique consists of several steps which are carried out one after the other and each one is a crucial step in the development phase. Some of these include requirements analysis, system design, implementation (coding), testing and maintenance. To ensure that there is a common understanding of the purpose of the product, the first step is collection of complete specifications. In the design phase, detailed designs of software components and system architectures are done. Subsequently, the Implementation phase follows where coders write code to the design specifications. After this stage, the development of the project is complete, and it goes into testing which involves checking whether or not the system behaves as intended and meets the quality criterion. Lastly, where the application is put in use over a specified period, in this case the maintenance stage provides for support, enhancements and bug fixes.

This linear method facilitates the development process through a set of procedures involving minimal risk since objectives and outputs are specified for each phase of the process.

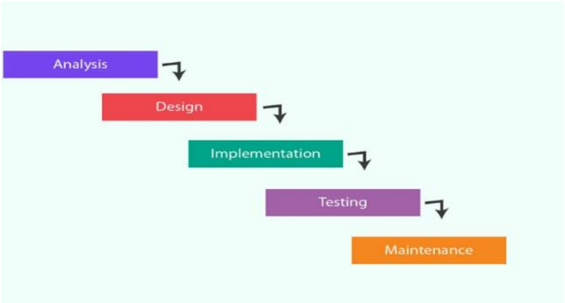

Fig.1 Waterfall Model

II. Literature Review

Waterfall methodology is one of the earliest organized approaches to software development and was founded by Winston W. Royce in the 70s to try and introduce some logic and order into the chaotic operations of the project. The Waterfall model is characterized by a linear & sequential development process and divides the project life cycle into clear stages including requirements analysis, design, implementation, testing and maintenance [7]. It is required to complete and confirm one step before proceeding with the next step. Due to this focus on defined measures and detailed documentation at every step, this model is very much documentation-driven and enhances the responsibility and traceability of the project task. The fixed nature of this phase-gate model has also gained its use in sectors such as manufacturing, military, and healthcare where there is a need for stability and excessive documentation with adherence to set regulations.

Studies highlight the merits of the Waterfall model for projects where the requirements are clear, and change is unlikely. Quite surprisingly, the model applications done by researchers of the University of Missouri-St Louis show that the model can also be applied in industries where there is strict requirement for documentation and quality assurance. Since the model is designed in stages, it is possible to determine the level of completion of each stage of the development and this helps to enhance predictability, control of the project and to minimize the chances of occurrence of changes that are not planned for. A sequential procedure often referred to as the Waterfall model is likely to promote a lengthy well-documented and feebly managed administration of the development process, which in some cases is needed due to the mandatory documentation and responsibility of all actions taken. As was shown in the case of the Old Dominion University study, this helps mitigate risks in the very first stages because there is extensive requirements and risks provision even before the design or coding starts. This approach and risk assessment in advance of project engagement is useful in industries in which stability and reliability of the outcome is more critical than the need for speed.

Even though there are these advantages, the weakness of the Waterfall technique that some people argue against is its rigidity which may lead to additional expenses and delays in case of changes in requirements during the project. Testing and validation phases are done at the end of the implementation stage, meaning that new features or bugs found towards the end of the development cycle may require a lot of reworking, often rewriting a good portion of previous phases. This phased approach to project development has been viewed as a serious limitation, especially for projects where the requirements are likely to evolve or are not fully known at the beginning [9]. In the words of the University of Houston-Clear Lake, there are constraints provoked by such feedback since a testing could only take place on the system as a whole, after the entire development work has been completed, which means that problems can only be found much later in the process, causing the issues to be more costly and time-consuming.

Given these limitations, several current software initiatives have adopted a more iterative approach called Agile, which highlights the significance of changes in client needs over time. Agile methodologies embrace an iterative approach and short sprints of deliverables to accommodate change, soliciting contact with the user owning the requirement helpfully throughout the project. Development methodologies such as Waterfall which encourage relaxed project environments that lengthen the period within which a project involves constant input from its clients tend to fall short in a managing project as system software development where there are practical objectives of speed and a constant incorporation of the client's input on the developing system. Graduate research conducted in ResearchGate shows that while Waterfall guarantees success for projects whose scope or boundaries do not shift, Agile is the most widely implemented in software development projects by far high as possible.

Thus, the use of the Waterfall model is getting more focused with it being applied only where requirements are stable, and compliance is a must. Research indicates that the characteristics of Waterfall concerning organized documentation and strict verification processes are useful in the regulation and production of drugs. Conversely, more often than not, agile methodologies are adopted for software development projects that require change hand in hand with the end users owing to the iterative mode of development. To summarize, it is clear that although projects that value a high degree of control through documentation and design have not made Waterfall redundant, its use is, more and more, confined to projects that have fixed requirements and where compliance is of utmost importance, with Agile becoming the most favored methodology in flexible and client-centered projects.

III. METHODOLOGY

The "Waterfall" approach divides software development into phases, with each phase providing input for the next. Fig.1 representation [6]

A. *Analysis Phase*

The behavior of the software that must be produced is fully described and is commonly referred to as a program Requirements Specification (SRS). System and business analysts are involved in defining the functional as well as non-functional requirements. Typically, use cases that describe how users interact with the software are used to develop functional requirements. They contain specifications like purpose, scope, viewpoint, functions, software attributes, user characteristics, functionalities, interface requirements, and database requirements. [1] On the other hand, rather than focusing on specific behaviors, the non-functional requirements refer to the different standards, limitations, and requirements placed on the software's operation and design. It has attributes like performance, quality standards, availability, maintainability, scalability, testability, and reliability.

B. *Design Phase*

It is a methodology for planning and resolving issues related to software solutions. It involves algorithm design, software architecture design, database conceptual schema and logical diagram design, idea design, graphical user interface design, and data structure definition to outline the plan for a solution.

C. Implementation Phase

This phase explains how programming and deployment into an executable program, database, website, or software component make business needs and design requirements a reality. Writing the real code, assembling it into a functional program, and producing the database and text files are all part of this step. [2] Otherwise, it is the process of integrating all designs and requirements into a production environment.

D. Testing Phase

Sometimes, it is referred to as validation and verification, and it is the process of ensuring that a software solution satisfies the initial requirements and specifications and fulfils the intended purpose. In fact, validation is the process of assessing software during or after the development process to figure out whether it satisfies predetermined requirements, whereas verification is the process of assessing software to figure out whether the products of a particular development phase satisfy the conditions imposed at the beginning of that phase. [3] Furthermore, the testing stage allows for debugging, which entails discovering, correcting, and resolving issues and system faults.

E. Maintenance Phase

It is the process of making changes to a software solution after it has been delivered and put into use to improve output, address defects, and improve quality and performance. [4] Additional maintenance tasks, such as enhancing program dependability, adapting to new user needs, and modifying the software to its environment, can also be completed during this phase.

IV. Result and Analysis

The Waterfall Model is the common software engineering and project management paradigm which executes activities in a linear sequential order from the beginning to the conclusion of a project. It consists of the stages of requirements, design, implementation, testing, and eventual upkeep of the developed software product. In this model, the successive phase can only be commenced after the current phase has been completed, tested and deemed fit for retention [8].

This model encourages good documentation and division of tasks which is a merit to industries that must deal with strict regulations and compliance such as the health and education where compliance requirements and audits are crucial [7]. Unfortunately, in this model, there can be disadvantages especially in those projects where flexibility is required since testing is at the tail end, and it becomes expensive to adjust after the work has proceeded to the implementation stage. Although the Waterfall model looks more appealing for those projects which are less prone to changes and have fixed requirements, Agile iterative models might as well suit best those projects which allow for flexibility [9].

V. Advantages and Disadvantages

Waterfall development has the benefit of enabling departmentalization and management. A timeline with due dates for every stage of development can be established and a product can proceed through the stages of the development process model one at a time. Analysis is the first step of development, which continues with design, implementation, testing, and maintenance. Every stage of the development process follows a set sequence. Though there is limited opportunity for reflection or modification in the waterfall development methodology. Once an application is in the testing stage, it is quite difficult to change something that was not well-documented or considered at the concept stage [8]. That is the main disadvantage. In addition, table 1 shown the main advantages and disadvantages. [5]

Advantages	Disadvantages
Straightforward and simple to use.	Until the end of the life cycle, no functional software is created.
The model's rigidity makes it easier to manage. Every phase has a review procedure and specific deliverables.	High levels of uncertainty and risk. This is because integration is done as "big bang" at the end, it is impossible to uncover any business or technology bottleneck or issues early on.
Phases are processed and finished simultaneously.	For complex and object-oriented tasks, this model is not suitable.
Works effectively for smaller projects with clearly defined requirements.	Poor model for projects that take a long time to complete.
Phases are well-defined, milestones are well-understood, activities are simple to	Not suitable for projects with moderate to high modify risk

organize, and the process and outcome are recorded.	requirements. Therefore, this process model has a high level of risk and uncertainly.

Table 1 Advantages and disadvantages of waterfall model

VI. Conclusion

To sum up, the Waterfall method's structured and systematic approach is highly beneficial for projects that have clear cut deliverables and deadlines, and where stability and detail, in terms of documentation, are a priority. This is ideal for the sectors which need to make detailed preparations beforehand since the execution of the project has clearly defined stages that can be controlled.

Nevertheless, it proves to be unfit for modernizations in the case when changes are oftentimes necessary and flexibility is in great demand, due to its rigidity and difficulty to implement changes. Thus, in today's business environment, clients prefer more iterative models, for instance, Agile, even though Waterfall is still applicable for some controlled situations as stable projects.

References

[1] Sommerville, *Software Engineering.* S.L.: Pearson Education India, 2016.

[2] U. Senarath, "Waterfall Methodology, Prototyping and Agile Development," 2021, doi: https://doi.org/10.13140/RG.2.2.17918.72001.

[3] *The Traditional Waterfall Approach*. (n.d.). https://umsl.edu/~hugheyd/is6840/waterfall.html

[4] Atlassian, B. (n.d.). *Waterfall Methodology for Project Management | Atlassian*. Atlassian. https://www.atlassian.com/agile/project-management/waterfall-methodology

[5] Radware bot manager Captcha, https://iopscience.iop.org/article/10.1088/1757-899X/1169/1/012025/pdf (accessed Oct. 23, 2024).

[6] R. S. Pressman and B. R. Maxim, *Software engineering: a practitioner's approach*. New York: Mcgraw-Hill Education, 2014.

[7] Zeil, S. (2013). *Software Development: The Waterfall Model*. https://www.cs.odu.edu/~zeil/cs333/sum13/Public/waterfall/waterfall-7-4x3.pdf

[8] Unknown. (n.d.). *The Software Development Life Cycle (SDLC) For Small to Medium Database Applications*. https://www.ecs.csun.edu/~rlingard/comp480/TheSoftwareDevelopmentLifeCycle.pdf

[9] Thesing, T., Feldmann, C., & Burchardt, M. (2021). Agile versus Waterfall Project Management: Decision Model for Selecting the Appropriate Approach to a Project. *Procedia Computer Science*, *181*, 746–756. https://doi.org/10.1016/j.procs.2021.01.227

Logbook

Date	Activity	Person In Charge
13/10/2024	Topic selection and brainstorming	All members
15/10/2024	Reviewed relevant papers using Google Scholar	All members
20/10/2024	Drafting abstract and introduction sections	Adriana
22/10/2024	Writing the methodology section	Ru Qian
23/10/2024	Drafting result and discussion sections	Ru Qian
25/10/2024	Finalizing the conclusion and references	Adriana
27/10/2024	Checking and formatting	All members
28/10/2024	Writing abstract and introduction sections	Adriana
29/10/2024	Writing literature review	Adriana
29/10/2024	Reviewing all sections	All members