Waterfall Model

# Abstract

Table of Contents

[Abstract 1](#_Toc88085856)

[Introduction 1](#_Toc88085857)

[Content 2](#_Toc88085858)

[History 2](#_Toc88085859)

[Features 3](#_Toc88085860)

[Phases 5](#_Toc88085861)

[Strength and weakness 6](#_Toc88085862)

[Comparison with other system methodology 7](#_Toc88085863)

[Factors that contribute us to choose the waterfall approach 7](#_Toc88085864)

[Conclusion 8](#_Toc88085865)

[References 8](#_Toc88085866)

[YouTube References 9](#_Toc88085867)

# Introduction

A system isa generalset of parts, steps, or components that are connected to form a more complex whole.Methodology definition is - a body of methods, rules, and postulates employed by a discipline: a particular procedure or set of procedures. System methodology together is a problem-solving method that involves looking at the wider system, breaking apart the parts, and figuring out how it works to achieve a particular goal. (European Journal of Preventive Cardiology, 2021)

The methodologies ensure a smooth software development experience while fulfilling project requirements. The software development methodology is concerned only with the planning aspects of the software development and hence it acts as a framework that is used to design, plan, and guide the software development process.

There are various types of system methodology namely:

* Waterfall model
* Agile model
* Incremental model
* Scrum

For this report, we are using references of Waterfall model to better understand it in depth. (Dutta)

Waterfall Model is a sequential model that divides software development into pre-defined phases. Each phase must be completed before the next phase can begin with no overlap between the phases. Each phase is designed for performing specific activity during the SDLC phase. It was introduced in 1970 by Winston Royce. The Waterfall Model is a linear application development model that uses rigid phases: When one phase ends, the next begins. Steps occur in sequence, and, if unmodified, the model does not allow developers to go back to previous steps (hence “waterfall”: Once water fall model, it cannot go back up).

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear**-**sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The Waterfall model is the earliest SDLC approach that was used for software development. The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap. (A., 2018)

# Content

## History

Waterfall model was primarily acknowledged for describing use of stages in software program engineering became held by Herbert D. Bennington at the Symposium on superior Programming techniques for virtual computers on 29 June 1956.This presentation was approximately the improvement of software for SAGE. In 1983 the paper became republished with a foreword via Bennington explaining that the stages were on motive organised in keeping with the specialisation of responsibilities and pointing out that the procedure changed into no longer in reality performed in a strict top-down style but relied on a prototype. (htt1)

Although the term "waterfall" is not used in the paper, the first formal certain diagram of the procedure later referred to as the "waterfall model" is regularly cited as a 1970 article by Winston W. Royce. but he additionally felt it had essential flaws stemming from the truth that checking out handiest befell on the cease of the technique, which he described as being "risky and invitations failure”. The relaxation of his paper added five steps which he felt had been necessary to "take away maximum of the development dangers" associated with the unaltered waterfall approach. (htt2)

Royce's five additional steps didn’t take mainstream hold, however his diagram of what he taken into consideration a mistaken procedure became the place to begin whilst describing a "waterfall" method.

The earliest use of the time "waterfall" may additionally have been in a 1976 paper by way of Bell and Thayer.

Later in 1985, the United States branch of protection captured this approach in DOD-STD-2167A, their standards for running with software improvement contractors, which stated that "the contractor shall implement a software improvement cycle that consists of the subsequent six levels: software Requirement evaluation, preliminary design, unique layout, Coding and Unit checking out, and integration checking out. At this juncture disclosed by means of Gains [7], Dr. Winston Royce wrote an editorial on handling massive and complex software developments in this newsletter, based totally on experiences in developing software program for planning spacecraft missions, commanding and submit-ﬂight analysis, W. Royce describes the fundamentals of software program/gadget improvement. Part of those basics are nevertheless applicable today. In an intermediate phase of his collaboration Roylee presents a sequence of phases of his elaboration Royce presents a sequence of phases which form a software development sequence, illustrated by Bell and Thayer will later refer to this sequence as the concept of the "waterfall" of development activities and will be the originators of the designation for the ﬁrst SDM, the waterfall model.

Additionally, Waterfall model began to have lots of issues therefore, Gains has the opinion the Waterfall Model predominately emphasizes on the freezing of requirement specifications or the high-level design very early in the development life cycle, prior to engaging in more thorough design and implementation work. So, the Waterfall model is likely to be unsuitable if requirements are

not well understood/defined or are likely to change in the course

of the project. Petersen associate the Waterfall Model

with high costs and efforts. They feel confirmed in their belief

by the number of documents to be approved in every phase, by

the difficulty to make changes, by the difficulty iterations take to

initiate and achieve and by problems that arise only in later phases.

Consequences of these issues have been that the customers’ current needs are not addressed, resulting in implemented but unused

features. To conclude his section, Mudassar and Govardhan

consider the Waterfall Model as a baseline for many -more

contemporary- Software Development Life-cycle Models, while

Dr. Winston Royce can be seen as the founding father of plan driven software development methodologies. (htt3)

## Features

Waterfall model being the root of all the other models, is still considered to be common in many other versions. It is also supported by many other methods and techniques. Waterfall model is also known best through its development feature for instance:

Steps in waterfall model are arranged in sequential order which means first of all, inputs are taken from previous step and outputs are given to the coming step after checking and matching all the entry criteria of the succeeding step. Waterfall model is best known to specify, design, build in sequence that is intuitively obvious and appears to be natural.

Furthermore, Waterfall model can also be distinguished with the help of advantages and disadvantages of that model.

Advantages of waterfall model:

* Waterfall model is simple to implement and the amount of resources

required for it are minimal.

* In this model, output is generated after each stage (as seen before), therefore it

has high visibility. The client and project manager gets a feel that there is

considerable progress. Here it is important to note that in any project psychological

factors also play an important role.

* Project management, both at internal level and client's level, is easy again

because of visible outputs after each phase. Deadlines can be set for the completion

of each phase and evaluation can be done from time to time, to check if project is going as per milestones.

* This methodology is significantly better than the haphazard approach to

develop software. It provides a template into which methods of analysis, design, coding, testing and maintenance can be placed.

* This methodology is preferred in projects where quality is more important as compared to schedule or cost. (htt4)

Disadvantages of waterfall model:

* Real projects rarely follow the sequential flow and iterations in this model are handled indirectly. These changes can cause confusion as the project proceeds.
* It is often difficult to get customer requirements explicitly. Thus specifications can't be freeze. If that case arises baseline approach is followed, wherein output of one phase is carried forward to next phase.
* In this model we freeze software and hardware. But as technology changes at a rapid pace, such freezing is not advisable especially in long-term projects.
* This method is especially bad in case client is not IT-literate as getting specifications from such a person is tough.
* Even a small change in any previous stage can cause big problem for subsequent phases as all phases are dependent on each-other. (htt5)

## Phases

In "The Waterfall" approach, the entire course of programming improvement is isolated into discrete stages. The result of one stage goes about as the contribution for the following stage successively. This implies that any stage in the improvement interaction starts provided that the past stage is finished. The cascade model is a consecutive plan process in which progress is viewed as streaming consistently downwards (like a cascade) through the periods of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation, and Maintenance. (Finlay, 2021)

As the Waterfall Model shows the product improvement process in a straight successive stream; consequently, it is likewise alluded to as a Linear-Sequential Life Cycle Model.

Different phases of the waterfall model are listed below:

Requirements: It gets all the necessities to plan and what is its capacity, reason, and so forth Here, the particulars of the information and yield or the eventual outcome are contemplated and checked. (Anderson D. , 2021)

System Design: The prerequisite determinations from the main stage are concentrated in this stage and framework configuration is ready. Framework Design helps in determining equipment and framework prerequisites and furthermore helps in characterizing by and large framework engineering. The product code to be written in the following stage is made at this point.

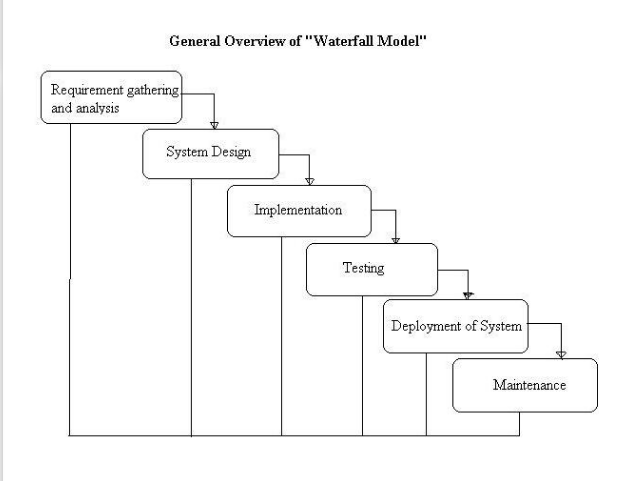
Implementation: With inputs from framework plan, the framework is first evolved in quite a while called units, which are coordinated into the following stage. Every unit is created and tried for its usefulness which is alluded to as Unit Testing.

Integration and Testing: All the units created in the execution stage are coordinated into a framework after testing of every unit. The product planned, necessities to go through steady programming testing to see whether there are any blemishes or mistakes. Testing is done as such that the customer doesn't deal with any issue during the establishment of the product.

Deployment of System: Once the useful and non-utilitarian testing is done, the item is sent in the client climate or delivered into the market.

Maintenance: This progression happens after establishment and includes adjusting the framework or a singular part to modify credits or further develop execution. These adjustments emerge either because of progress demands started by the client, or imperfections revealed during live utilization of the framework. The customer is furnished with ordinary upkeep and backing for the created programming.

This large number of stages are fell to one another in which progress is viewed as streaming consistently downwards (like a cascade) through the stages. The following stage is begun solely after the characterized put forward of objectives is accomplished for the past stage and it is closed, so the name is given Waterfall Model.

*Fig 1.1 Different phases of waterfall model.* (htt6)

We can also take reference from the diagram above. It helps to understand how waterfall model uses different phases to work appropriately. Briefly explaining, Waterfall model gathers all the necessary requirements needed before designing. Then it moves forward to implementation phase. It then tests before deploying the system. After everything is done, it looks after the maintenance.

## Strength and weakness

Waterfall model is followed by its unique strengths and weaknesses. Although it is convenient and easy to understand, some critics often say it does not provide best flexibility. For better understanding, few strengths and weaknesses of waterfall model are listed as below: (Petersen & Wohlin, 2004)

STRENGTHS:

* Easy to understand and easy to use.
* Provides a reference to inexperienced staff.
* Milestones are well-understood by the team.
* Provides requirements stability.
* Facilitates strong management control.
* Works well when quality is more important than cost or schedule.

WEAKNESSES:

* All requirements are to be known upfront.
* Deliverables are created for each phase and considered to be frozen – inhibits flexibility.
* Can give a false impression of progress.
* Integration is one big bang at the end.
* Little opportunity for customer to preview the system. (Tutorials point )

## Comparison with other system methodology

(Casteren)

In comparison to other system methodology Waterfall model is one of the best development models that have been introduced in the software development circles. It is also known as the equivalent as the model implies in each stage to be completed before going to the next one. So, the model goes continuously step by step without having any difficulties in development stage. Waterfall model resembles a real-life waterfall that has a beginning and an end with water falling freely from the cliff without being interrupted or altered. The Waterfall model is applied to the software development life cycle when it is most suitable for a certain project. (Kannan V., 2014)

(Kisielnicki J., 2017)

## Factors that contribute us to choose the waterfall approach

* Short-term projects.
* Clear, fixed, and well-documented requirements.
* Static technology that is understood.
* Stable product definition.
* A variety of resources for product support.
* Low Product Owner Involvement.
* Strictly set timeline.
* Fixed budget that cannot be changed.

Comparing to other system methodology, Waterfall is that the model is in strict order with regards to the tasks carried out and their timelines. Due to being fixed, the tasks can be easily divided between software departments and monitored as to task completion. Clients may find the Waterfall Model a reliable approach as they will be aware of the whole process from the start till finish beforehand. As for the disadvantages of the Waterfall Model, there are some as well. For instance, Waterfall has no revision of a task. The farther the application development goes forward into the other stage the harder it is to go back and change its conceptual basics or documentation. (Anderson D. , 2004)

# Conclusion

# References

(n.d.). Retrieved from https://www.ukessays.com/essays/information-technology/the-history-of-the-waterfall-model-information-technology-essay.php

(n.d.). Retrieved from https://airbrake.io/blog/sdlc/waterfall-model

(n.d.). Retrieved from https://en.wikipedia.org/wiki/Waterfall\_model

(n.d.). Retrieved from https://www.idc-online.com/technical\_references/pdfs/data\_communications/Waterfall\_Model\_Advantages\_and\_Disadvantages.pdf

(n.d.). Retrieved from https://magadhuniversity.ac.in/download/econtent/pdf/What%20is%20Waterfall%20Model%20in%20SDLC-converted%20(1).pdf

(n.d.). Retrieved from https://www.tutorialspoint.com/sdlc/sdlc\_waterfall\_model.htm

A., D. A. (2018). From waterfall to agile software: Development models in IT sector. *Impacts on company management. Journal of international* , 315-326.

Anderson, D. (2004). Agile Management for Software Engineering. In *Applying the Theory of Constraints for Business Results.*

Anderson, D. (2021). Waterfall Model . *Science Direct* .

Casteren, W. (n.d.). The Waterfall Model and Agile Methodologies :.

Dutta, G. (n.d.). Waterfall Model. In *www.academia.edu.*

*European Journal of Preventive Cardiology*. (2021). Retrieved from Standard journal publication model : (https://academic.oup.com/journals/pages/open\_access/funder\_policies/chorus/standard\_publication\_model)

Finlay, E. (2021, June 29). 5 waterfall project management phases you should know about. *Mind Manager Blog*.

Kannan V., J. S. (2014). Agile vs waterfall: A comparative analysis. *International journal of science , engineering and technology research*, 10.

Kisielnicki J., M. M. (2017). Effectiveness of agile compared to waterfall implementation . *Foundations of management* , 271-284.

Petersen, K., & Wohlin, ,. C. (2004). The Waterfall Model in Large-Scale Development. In *Software Engineering, 7th edn.*

Tutorials point . (n.d.). In *SDLC- Waterfall model.*

Y., B. (2012). A simulation model for the waterfall software development life cycle, . *International journal of engineering and technology.*, pp. 1.

# YouTube References

<https://youtu.be/6-GbY4Asgf8>

<https://youtu.be/1W8ZYtCPgYY>

<https://youtu.be/WmuUSKOUVHA>

<https://youtu.be/yryJ-6Ck4Xk>

<https://youtu.be/5RocT_OdQcA>