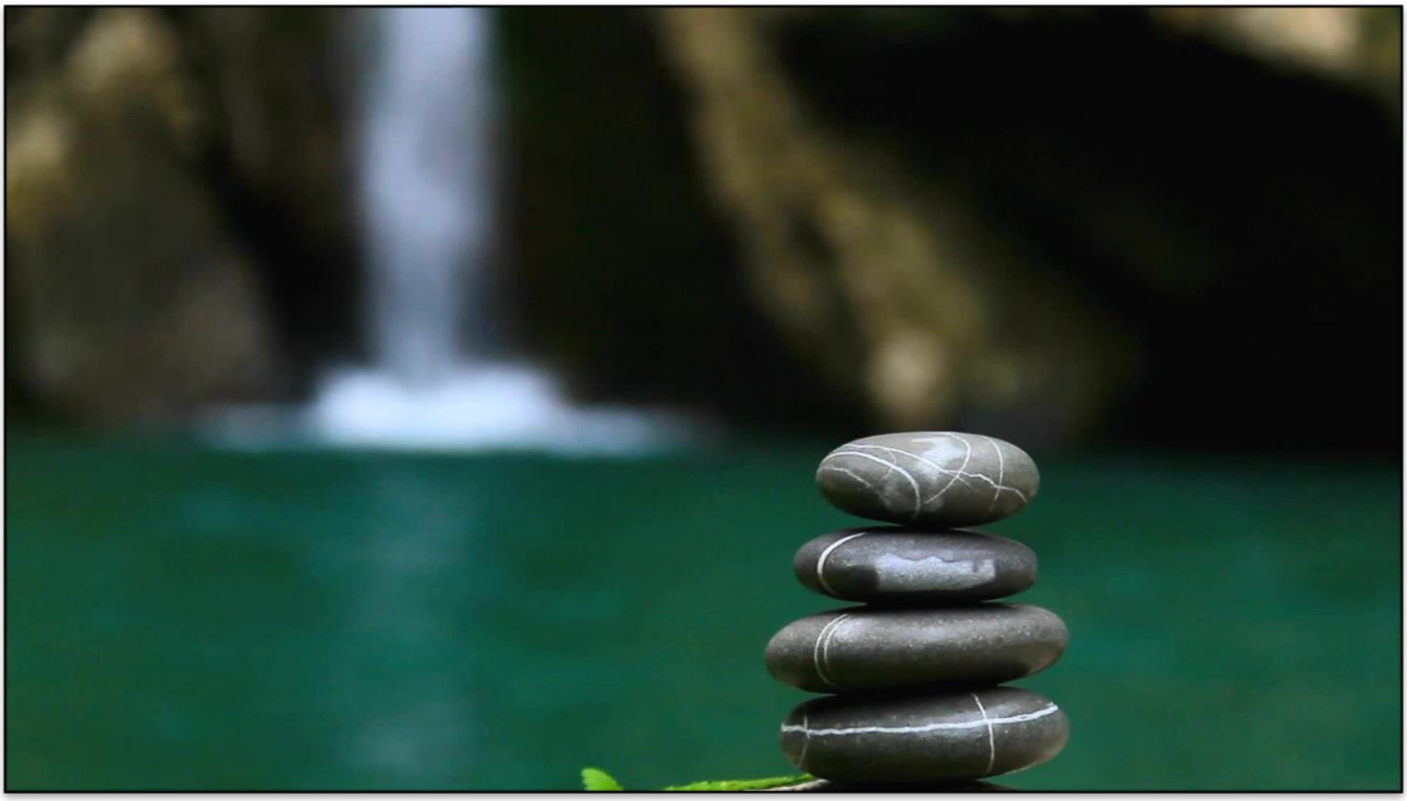
Waterfall Model

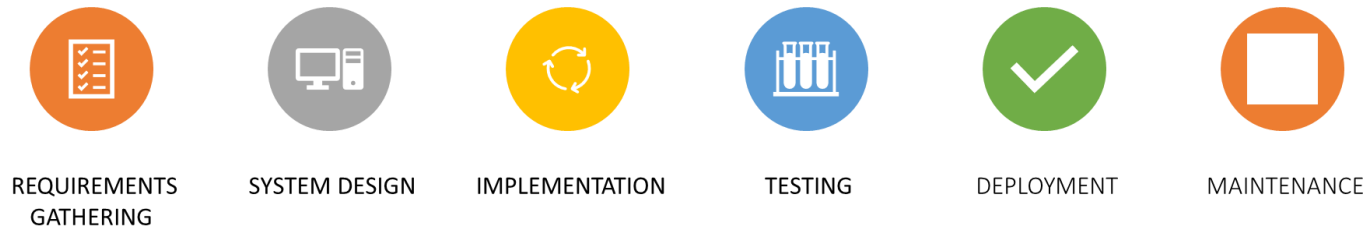


What is Waterfall Model?

The waterfall model is the first explicit model for software development process and was derived from other Engineering Processes in 1970's. This is a breakdown of project activities into linear sequential phases, meaning they are passed down onto each other, where each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks. Because of the cascade from one phase to another, this model is known as the "Waterfall Model".

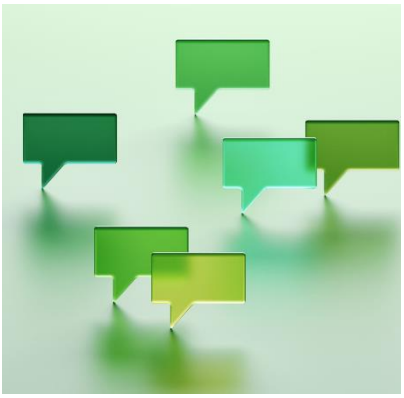
There are six main phases of waterfall model.

- Requirement Gathering
- System Design
- Implementation
- Testing
- Deployment
- Maintenance



(01) Requirement Gathering

The first step in the waterfall model is to gather and analyze the requirements for the software system. This involves meeting with stakeholders, such as users, managers, and customers, to understand their needs and expectations. The requirements should be clear, complete, and consistent. Once the requirements have been gathered, they should be documented in a requirements specification document. This document should describe all the requirements for the system in detail.

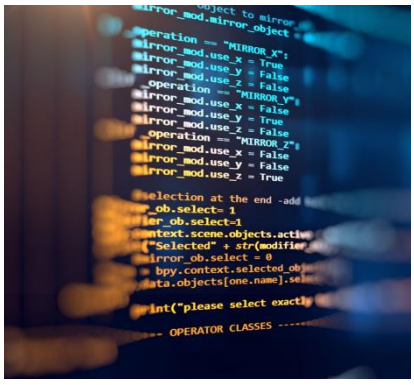


(02) System Design

The next step is to design the system architecture. This involves identifying the major components of the system and how they will interact with each other. The design should be efficient, scalable, and maintainable. Once the design is complete, it should be reviewed with the stakeholders to ensure that it meets their requirements.

(03) Implementation

The implementation phase involves developing the software according to the system design. The code should be well-structured, modular, and reusable. Unit testing should be performed on each component as it is implemented. Once all of the components have been implemented, integration testing should be performed to test the system.

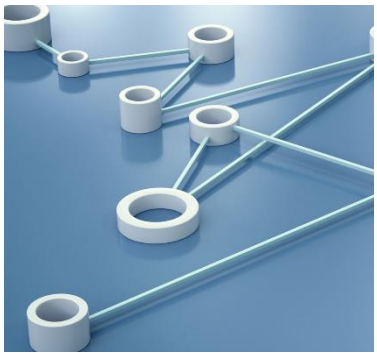


(04) Testing

The testing phase involves testing the software to ensure that it meets the requirements and is free of defects. This includes unit testing, integration testing, system testing, and acceptance testing. Unit testing is performed on each component individually to ensure that it works as expected. Integration testing is performed to test the interactions between the components. System testing is performed to test the system. Acceptance testing is performed by the stakeholders to ensure that the system meets their requirements.

(05) Deployment

The deployment phase involves installing the software on the production environment. This should be done carefully according to a deployment plan. The system should be monitored after deployment to ensure that it is working properly.



(06) Maintenance

The maintenance phase involves fixing bugs and adding new features to the software. It is important to maintain the software after deployment to ensure that it remains secure and meets the needs of the users.

Okay, now we all know what the waterfall model is and its phases. So, let's dive into our point of view to look at how we did it.

(01) Requirement Gathering

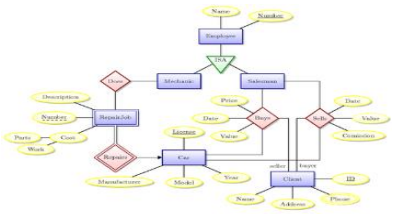
We did research on existing Learning Management Systems and reviewed existing documentation. By that we got to know so many things including,

- What are the Requirements need to be added?
- What are the problems that users face?
- Why they are very complex to use?

Then we stepped into our second phase.

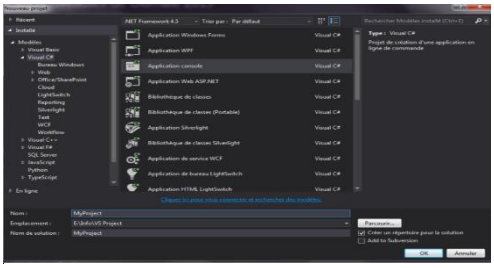
(02) System Design

We used Flow Charts, Use-case diagrams, Entity-Relationship Diagrams to create a high-level architecture of the system. We also created detailed designs of the individual components using a variety of tools and techniques. First, we created an Overview document of our system, which shows how every form is going to look like in our system. By these things it was easy to step into the next phase.



(03) Implementation

We used C# programming language and tools to develop the software system. We also wrote unit tests to ensure that the individual components of the system were working correctly. Not only that but also, we added so much new ideas while implementing the system, then that becomes more user-friendly.



(04) Testing

We used a variety of testing tools and techniques to test the software system. We also involved ourselves in the testing process to ensure that the system met our goals. As students we have used number of Learning Management Systems, so, we had a good idea how it needs to be created.

(05) Deployment

We are planning to deploy the software system to the cloud using a variety of tools and techniques. We also created user manuals and other documentation to support the system.

(06) Maintenance

We are planning to maintain our LMS by adding many more user-friendly options and to keep a good connection with our users to always give them the best experience.

The waterfall model is a classic software development methodology that has been used for decades. It is a sequential approach, where each phase of the development process must be completed before the next phase can begin.

The waterfall model has several advantages, including:

- It is simple and easy to understand.
- It provides a clear structure for the development process.
- It is well-suited for projects with well-defined requirements.
- It can help to ensure that projects are completed on time and within budget.

However, the waterfall model also has a number of disadvantages, including:

- It is inflexible and does not allow for changes to requirements once the project has begun.
- It can be risky for large and complex projects, as it is difficult to identify and address all of the requirements upfront.
- It can lead to the development of systems that do not meet the needs of the users, as the users are not involved in the development process until the later stages.

So, is the waterfall model a good choice for your software development project? It depends.

If your project has well-defined requirements and you are confident that the requirements will not change, then the waterfall model can be a good choice. However, if your project is large and complex, or if you are not sure what the requirements will be, then you may want to consider using a different software development methodology.

Ultimately, the best way to decide whether or not to use the waterfall model is to consider the specific needs of your software development project.

References:

- *Waterfall model*
  https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model

- *The Waterfall Model Software Development Methodology: A Comprehensive Guide*
  https://appmaster.io/blog/waterfall-methodology

- *The pros and cons of Waterfall methodology*
  https://www.lucidchart.com/blog/pros-and-cons-of-waterfall-methodology

- *Referred You-Tube Links,*
  https://youtu.be/bNLcRdrSQAU
  https://youtu.be/5RocT_OdQcA

- *Esoft Metro Campus – DiTEC Text book part ii*