Waterfall model

It is the first approach and the basic model used in software development. It is a simple model that is easy to use as well as understand. The execution happens in the sequence order, which means that the outcome of the one-stage is equal to the input of another stage. That's why it is also known as the Linear-sequential life cycle model.

To avoid the overlapping issues of the multiple phases, every stage should be completed before moving to the next stage. Each stage of the waterfall model involves the deliverable of the previous stage, like requirements, are transferred to the design phase, design moved to development, and so on. When we have the Life critical (hospital application) and Machine critical (Military project), we will widely use the waterfall model.

The waterfall model is divided into various stages, which are as follows:

* **Requirement collection**
* **Feasibility study**
* **Design**
* **Coding**
* **Testing**
* **Installation**
* **Maintenance**

Let us understand them one by one:

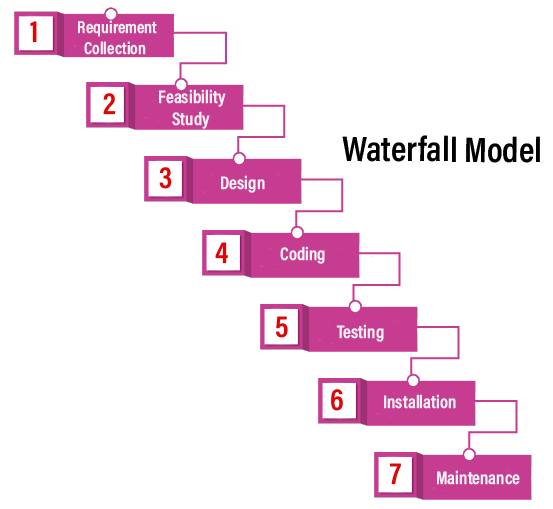
### Requirement Collection

Requirement collection is the first phase of the waterfall model, where a business analyst will assemble all the information or business needs of the client in the form of a requirement document. And this document should be clear and easy to understand, and all requirements are correctly listed.

By taking help of Software Requirement Specification [SRS], Customer Requirement Specification [CRS], and Business Requirement Specification [BRS], the SRS document is generated. And this SRS document covers the whole thing that should be developed and designed.

### Features of a functional requirement

* It should be written in a simple language so it can be easily understandable.
* The specification should be in the proper flow.
* The requirement should be countable.



### Feasibility Study

The feasibility study is based on the needs of the project, where many people (human resource, business analyst, architecture) evaluate whether the project can be done or not. To develop a good project, we should follow the various characteristics, which are based on the customer requirements:

|  |  |
| --- | --- |
| **Aspects** | **Description** |
| **Legal** | Can the company handle the project as cyber law and other monitoring agreements? |
| **Technical** | Check whether the available machine supports the software or not? |
| **Operation feasibility** | The company should able to generate operation that is given by the clients? |
| **Economic** | Should the company able to complete the product within given  budget or not? |
| **Schedule** | The project should be done within the given schedule or not. |

### Design

Once we are done with the feasibility study, we will move to our next stage, which is designing. In this, we will create the architecture of the product, with the help of some essential tools like a combination of different software and hardware, various programming languages (PHP, Java, .Net, etc.), database (MySQL, Oracle). And then the designer gets ready with a plan for the application that could be classified into two different parts:

* **High-Level Design**
* **Low-Level Design**

**High-Level Design [HLD]:**

In this, the designer will concentrate only on the models such as decision trees, flow diagrams, decision tables, flow charts, data dictionary, and the architect does it.

**Low-Level Design [LLD]:**

In this, the designer will concentrate on the components like a User interface (UI), and the developer manager does it.

### Coding

Once we are done with the design stage, we are ready to develop the application. For this, the developer will start writing the code based on their programming language knowledge, and it could be any language such as Python, C, Java, C#, C++, and so on. Whereas the back-end developers will do the back-end coding based on the needed operations, and the front-end developers will develop the attractive GUI.

### Testing

After the compilation of coding, it will hand over to the concern test engineer. And after that, the test engineer will start testing the functionality of the application based on the client's requirement.

While testing the application, they may encounter some defects or bugs (not working as per the client's needs) in the application and send those bugs to the developer with the proper justification. And the developer will verify that the given bug is valid or not. If it is correct, it will be fixed by the developer and change with the new one. After that tester will re-test it and verify that the bug is fixed or not.

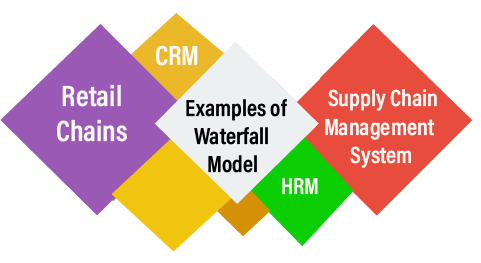
### Installation

Once the application is tested, we will move to the next stage (installation). In this, the process will remain until the software is stable or bug-free and fulfilling all the customer requirements. When the application is stable, it will install into the client's environment for their use.

After getting the software, the client will perform one round of testing to their satisfaction. If they face any error, they will inform the development team to resolve those issues for the particular application. When all the issue gets resolved, the application will be deployed for the end-users use.

## Example of the waterfall model

Earlier it was used for the applications such as **Human Resource Management [HRM], Supply Chain Management System, Customer Relationship Management [CRM], and Retail Chains**, etc. but now in present time, the waterfall models are replaced by other models such as **Iterative models and Agile Methodology**, etc.



## Pros and Cons of the Waterfall Model

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| In the Waterfall model, the requirement should be clear. | This model has no parallel deliverable, which means that two teams can work together. |
| It is suitable for a smaller project where needs are well understood. | The waterfall model doesn't provide the requirement changes and requirement review. |
| This model is easy to understand, as well as easy to use. | Previously, when the waterfall is invented, there is no concept of testing, that's why the developer is used to test the application. |
| It will allow us to arrange the tasks efficiently. | In between, changes are not allowed because one phase is dependent on another stage. |
| In this model, release level changes are allowed. | Backward tracking is not possible. |
| In this model, the procedure and the results are well documented. | It is a time-consuming process. |