Agile

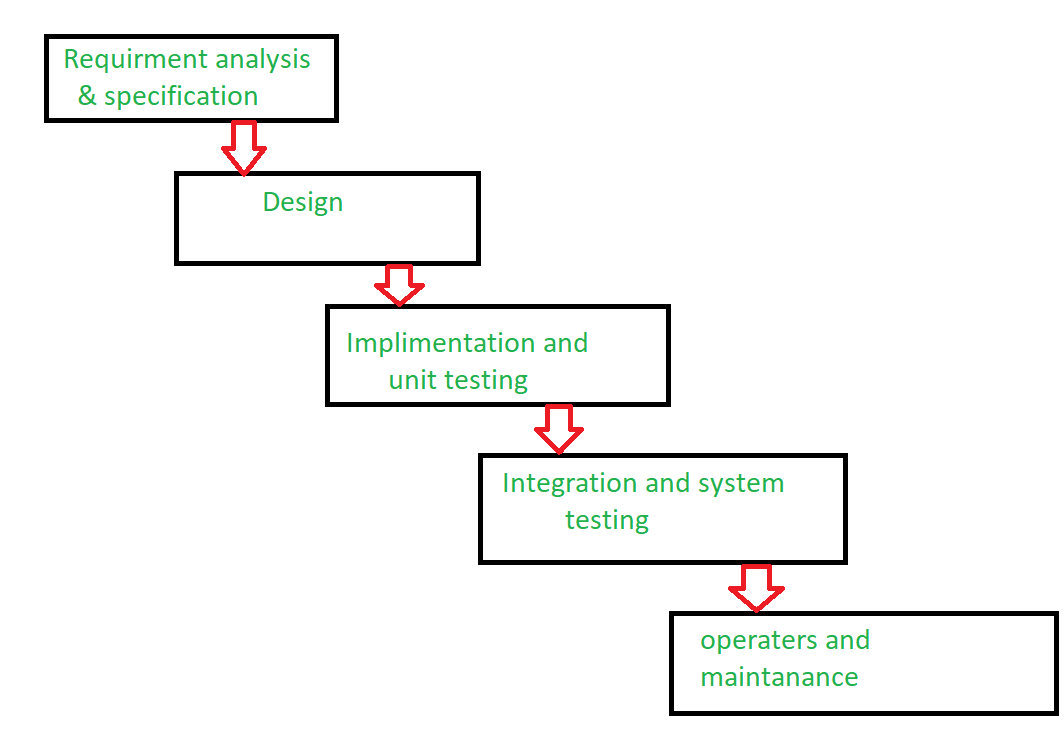
Failure of waterfall model:

## What is the Waterfall Model?

The [Waterfall Model](https://www.geeksforgeeks.org/waterfall-model/) is also known as the traditional waterfall software life cycle model. It is very simple to understand and use.

1. In the waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.
2. The classical waterfall model divides the life cycle into a set of phases.
3. The waterfall model considers that one phase can be started after completion of the previous phase, which is the output of one phase will be the input to the next phase.
4. Due to this the development process can be considered as a sequential flow in the waterfall. Here the phases do not overlap with each other.

This model divides the life cycle of a software development process into phases as shown below:

Waterfall Model

## Reasons for Failure of Waterfall Model

The traditional waterfall model suffers from various shortcomings, basically we can’t use it in real projects, but we use other software development lifecycle models which are based on the classical waterfall model. There are some reasons which are given below due to this waterfall model fails.

### 1. One Way Street

This model is just like the one-way street. Once phase X is completed and next phase Y has started then there is no way to going back on the previous phase. This is one of the issues to the failure of the waterfall model.

### 2. Overlapping

The waterfall model has lacked an overlapping among phase.The waterfall model recommends that new phase can start only after the completion of the previous phase. But in real projects, this can’t be maintained. To increase the efficiency and reduce the cost, phases may overlap.

### 3. Interaction

The waterfall model has lacked interaction among phase. Users have little interaction with project them. This feedback is not taken during development. After a development process starts, changes can not accommodate easily.

### 4. Support Delivery of System

The waterfall model does not support delivery of system in pieces. After a development process starts, changes cannot accommodate easily.

### 5. Feedback Path

The waterfall model has no feedback path. In the traditional waterfall model evolution of software from one phase to another phase is like a waterfall. The waterfall model assumes that no error is ever committed by developers during any phases. Hence, it does not incorporate any mechanism for error correction.

### 6. Not Flexible

Difficult to accommodate change requests. The waterfall model assumes that all the customer requirements can be completely and correctly defined at the beginning of the project, but actually customers’ requirements keep on changing with time. After the requirements specification phase is completed difficult to accommodate any change requests.

So, we can’t use real time projects in this model ,so we go for agile methodology.

Agile:

"**Agile process model**" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.



Phases of Agile Model:

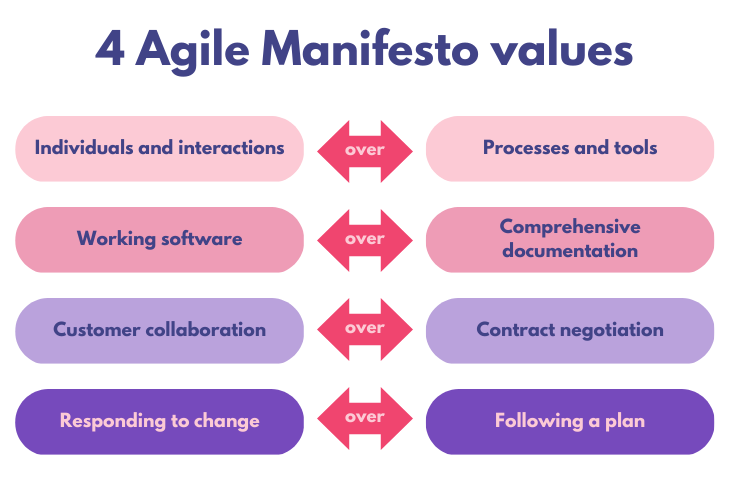
Following are the phases in the Agile model are as follows:

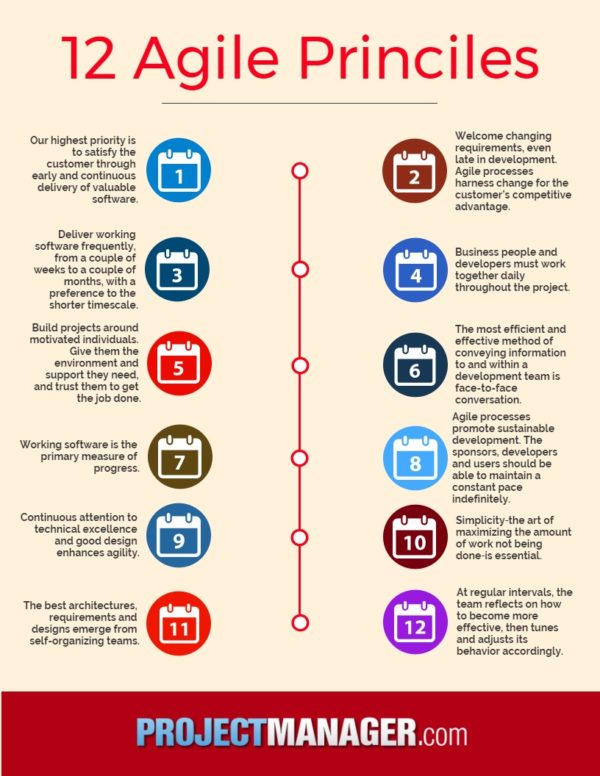
1. Requirements gathering
2. Design the requirements
3. Construction/ iteration
4. Testing/ Quality assurance
5. Deployment
6. Feedback
7. **Requirements gathering:** In this phase, you must define the requirements. You should explain business opportunities and plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.
8. **Design the requirements:** When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.
9. **3. Construction/ iteration:** When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.
10. **4. Testing:** In this phase, the Quality Assurance team examines the product's performance and looks for the bug.
11. **5. Deployment:** In this phase, the team issues a product for the user's work environment.
12. **6. Feedback:** After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.

Agile testing methods:

* Scrum
* Crystal
* Dynamic Software Development Method(DSDM)
* Feature Driven Development(FDD)
* Lean Software Development
* eXtreme Programming(XP)

4 values of agile:





Links for agile:

a. https://agilemanifesto.org/

b. Link for Agile Workflow :

<https://www.atlassian.com/agile/project-management/workflow>

c. Agile presentation attached

d. Agile retrospective :

<https://www.aha.io/roadmapping/guide/agile/what-is-an-agile-retrospective>

e. Agile By Example [video] :

<https://www.youtube.com/watch?v=o_4z-dfA6PY>