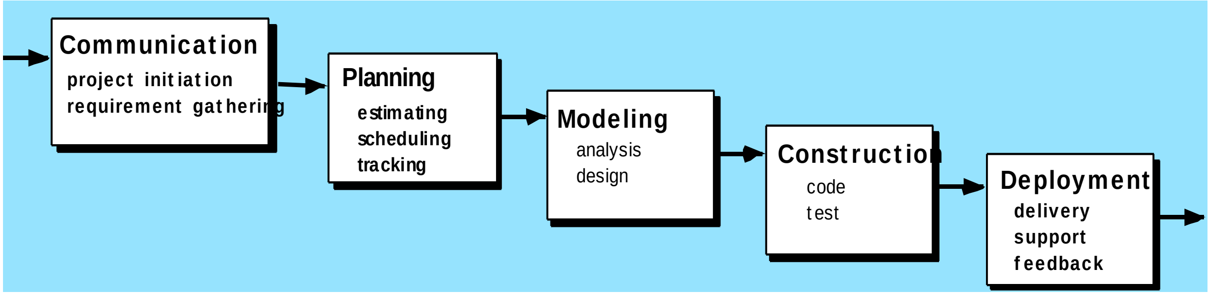
Waterfall Model

This model is also called as the **classic life cycle model** as it suggests a systematic sequential approach to software development.



5 Phases Of Waterfall Model :

1. Communication
2. Planning
3. Modeling
4. Construction
5. Deployment

1. Communication

In communication phase the major task performed is requirement gathering which helps in finding out exact need of customer. Once all the needs of the customer are gathered the next step is planning.

2. Planning

In planning major activities like planning for schedule , keeping tracks on the processes and the estimation related to the project are done. Planning is even used to find the types of risks involved throughout the projects. Planning describes how technical tasks are going to take place and what resources are needed and how to use them.

3. Modeling

This is one the important phases as the architecture of the system is designed in this phase. Analysis is carried out and depending on the analysis a software model is designed. Different models for developing software are created depending on the requirements gathered in the first phase and the planning done in the second phase.

4. Construction

The actual coding of the software is done in this phase . This coding is done on the basis of the model designed in the modeling phase. So in this phase software is actually developed and tested.

5. Deployment

In this last phase the product is actually rolled out or delivered & installed at customer's end and support is given if required. A feedback is taken from the customer to ensure the quality of the product.

When to use the waterfall model:

- This model is used only when the requirements are very well known, clear and fixed.
- Product definition is stable.
- Technology is understood.
- There are no ambiguous requirements
- Ample resources with required expertise are available freely
- The project is short.

Advantages Of Waterfall Model

1. Being a linear model, it is very simple to implement.The amount of resources required to implement this model are minimal.
2. Documentation is produced at every stage of the software's development. This makes understanding the product designing procedure, simpler.
3. After every major stage of software coding, testing is done to check the correct running of the code.

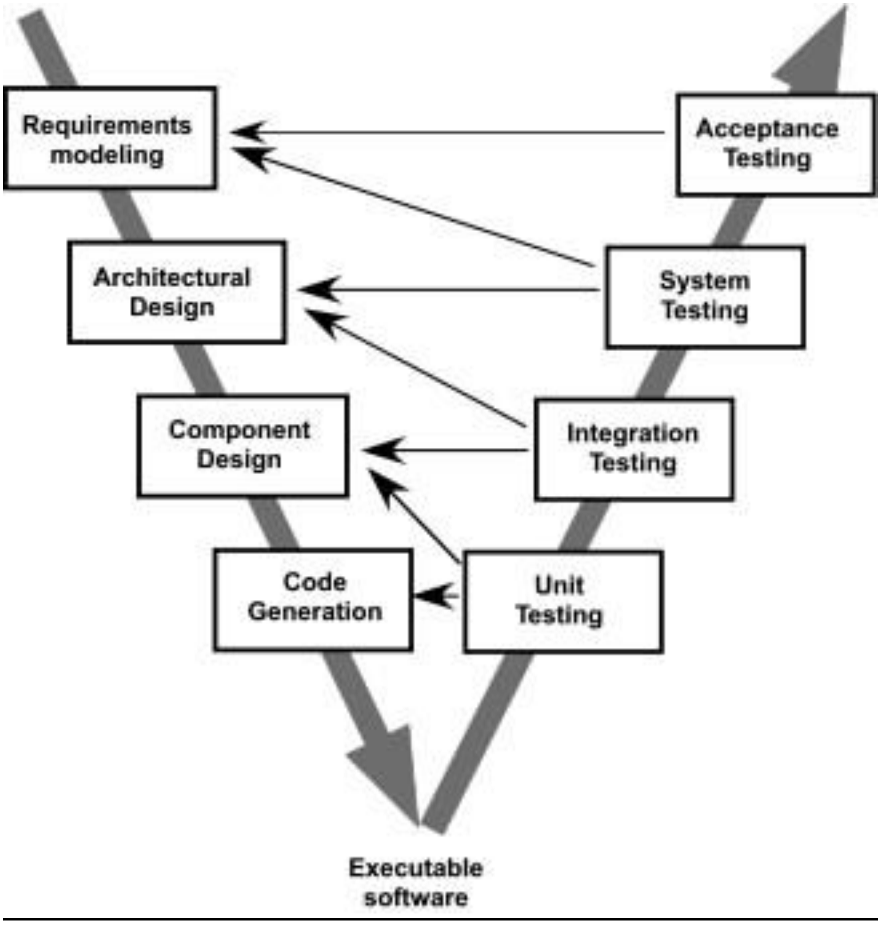
Disadvantages of Waterfall Model

1. Ironically, the biggest disadvantage is one of its greatest advantages. You cannot go back a step; if the design phase has gone wrong, things can get very complicated in the implementation phase.

2. Often, the client is not very clear of what he exactly wants from the software. Any changes that he mentions in between, may cause a lot of confusion.
3. Small changes or errors that arise in the completed software may cause a lot of problems.
4. Until the final stage of the development cycle is complete, a working model of the software does not lie in the hands of the client. Thus, he is hardly in a position to inform the developers, if what has been designed is exactly what he had asked for.

The V-Model

V- model means Verification and Validation model. Just like the waterfall model, the V-Shaped life cycle is a sequential path of execution of processes. Each phase must be completed before the next phase begins. Testing of the product is planned in parallel with a corresponding phase of development in **V-model**.



Unit testing: Testing by analysis of the code by developers for their independent modules is done.

Integration testing: Independent modules are tested together to validate interface and expose errors in them.

System testing: The system is tested against the system specifications.

User Acceptance testing: Testing is performed by end users to validate that the requirements mentioned in requirements phase have been met by the system or not before accepting it for production.

When to use the V-model:

- The V-shaped model should be used for small to medium sized projects where requirements are clearly defined and fixed.
- The V-Shaped model should be chosen when ample technical resources are available with needed technical expertise.

Advantages of V-model:

- Simple and easy to use.
- Testing activities like planning, test designing happens well before coding. This saves a lot of time. Hence higher chance of success over the waterfall model.
- Proactive defect tracking – that is defects are found at early stage.
- Avoids the downward flow of the defects.
- Works well for small projects where requirements are easily understood.

Disadvantages of V-model:

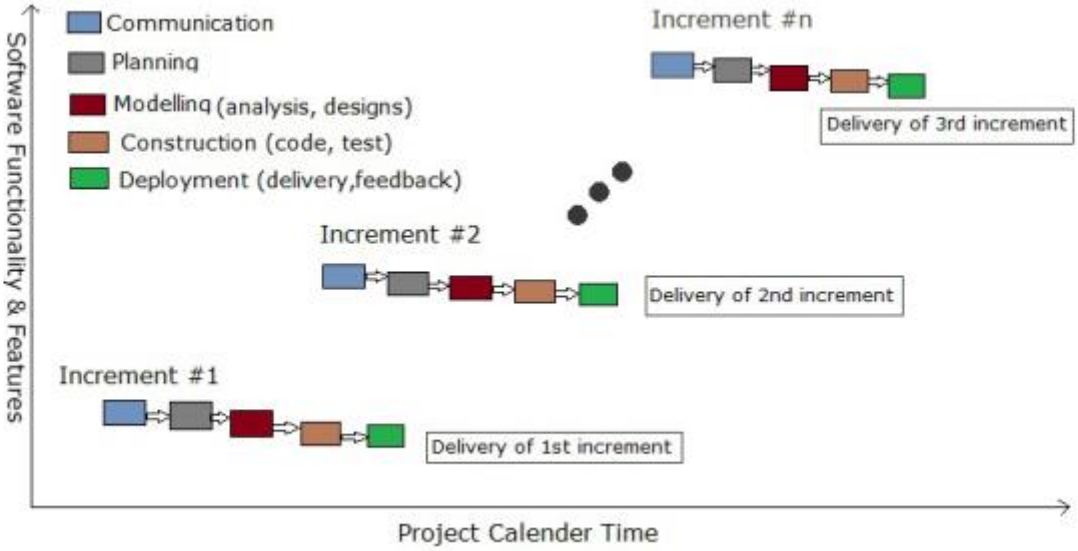
- Very rigid and least flexible.
- Software is developed during the implementation phase, so no early prototypes of the software are produced.
- If any changes happen in midway, then the test documents along with requirement documents has to be updated.

Incremental Model

Incremental model in software engineering is a one which combines the elements of waterfall model which are then applied in an iterative manner. It basically delivers a series of releases called increments which provide progressively more functionality for the client as each increment is delivered.

In incremental model of software engineering, waterfall model is repeatedly applied in each increment. The incremental model applies linear sequences in a required pattern as calendar time passes. Each linear sequence produces an increment in the work.

Diagram Of Incremental Model


Communication
Planning
Modelling (analysis, designs)
Construction (code, test)
Deployment (delivery,feedback)
Increment #n
Delivery of 3rd increment
Increment #2
Delivery of 2nd increment
Increment #1
Delivery of 1st increment
Software Functionality & Features
Project Calender Time

The first increment is often a core product where the basic requirements are addressed and the supplementary features are added in the next increments. The core product is used and evaluated by the client. Once the core product is evaluated by the client there is plan development for the next increment. Thus in every increment the needs of the client are kept in mind and more features and functions are added and the core product is updated. This process continues till the complete product is produced.

The increments earlier to the main increment are called as “stripped down” versions of the final product. These increments form a base for customer evaluation. On this basis client can suggest new requirements if required.

If there are less number of employees to work on the project Incremental development model is very useful to complete the project before the deadline. In a project early increments can be done with less number of people. In case if the core product is well-defined and understood more employees can be added if needed in the future increments.

One of the benefits of Incremental process model is that it can be planned to manage technical risks.

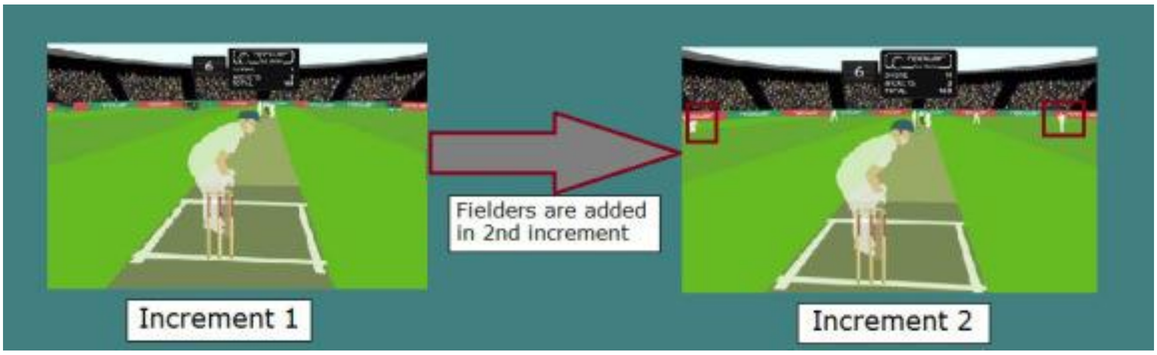
Advantages Of Incremental Model

- Initial product delivery is faster.
- Lower initial delivery cost.
- Core product is developed first i.e main functionality is added in the first increment.
- After each iteration, regression testing should be conducted. During this testing, faulty elements of the software can be quickly identified because few changes are made within any single iteration.
- It is generally easier to test and debug than other methods of software development because relatively smaller changes are made during each iteration. This allows for more targeted and rigorous testing of each element within the overall product.
- With each release a new feature is added to the product.
- Customer can respond to feature and review the product.
- Risk of changing requirement is reduced
- Work load is less.

Disadvantages Of Incremental Model

- Requires good analysis.
- Resulting cost may exceed the cost of the organization.
- Each phase of an iteration is rigid and do not overlap each other.
- As additional functionality is added to the product, problems may arise related to system architecture which were not evident in earlier prototypes.

Example of Incremental Model :



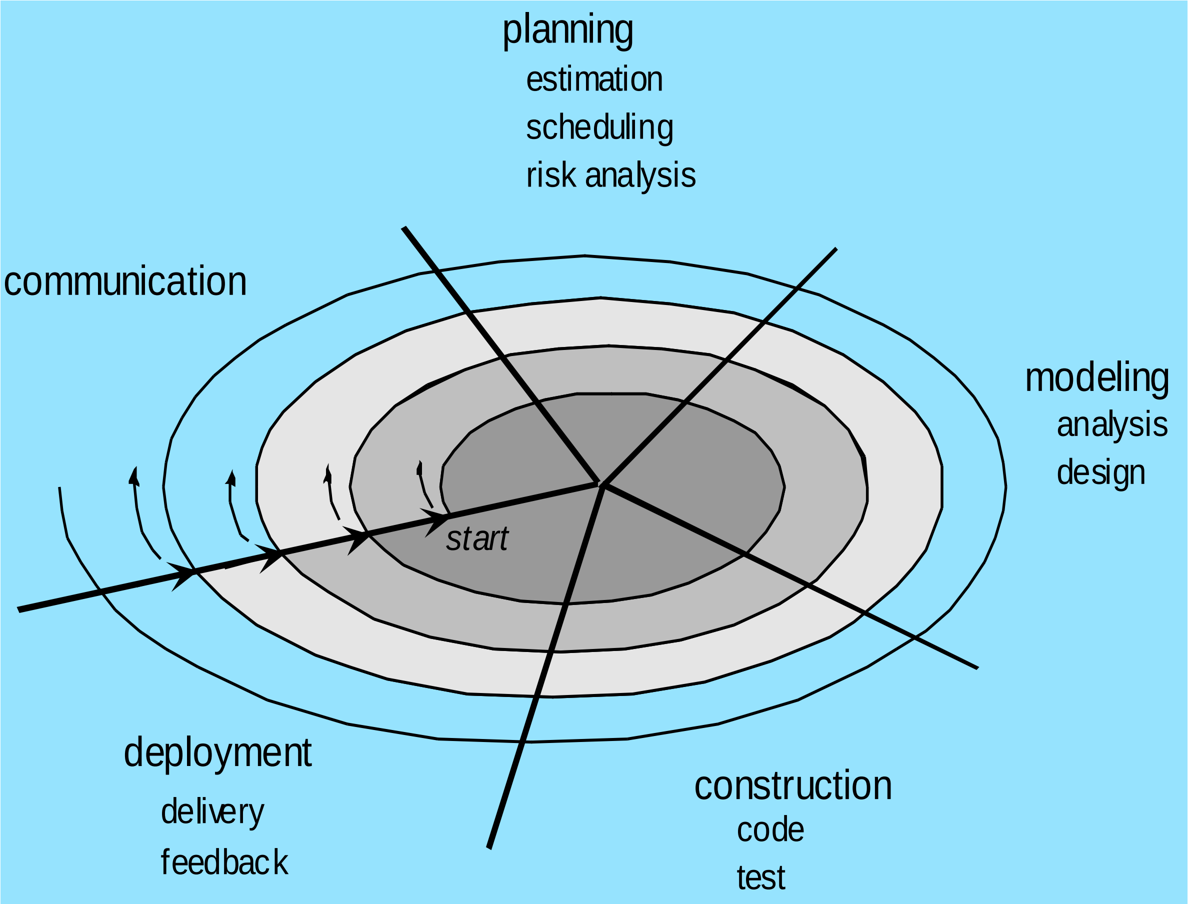
As you can see in the above example the core product is a cricket game where only 2 necessary players and one umpire is present in the game. In the next increment fielders are also added in the game and thus the increments goes on with new features being added with every increment.

Spiral Model

Spiral model is an evolutionary software process model which is a combination of an iterative nature of prototyping and controlled and systematic aspects of traditional waterfall model.

In Spiral model, software development takes place in series of developed releases. In initial stage iterations, the release or model might be a paper model or a prototype. In the later stages a more complete version of software is actually produced.

A spiral model is divided into number of framework activities, also called task regions. Every framework activities represent one section of the spiral path. As the development process starts, the software team perform activities that are indirect by a path around the spiral model in a clockwise direction. It begins at the center of spiral model.



Phases Of Spiral Model (Tasks Regions)

- Customer communication-tasks required to establish effective communication between developer and customer.
- Planning-tasks required to define resources,timelines and other project related information.
- Risk analysis tasks required to assess both technical and management risks.

- Construction and release tasks required to construct ,test,install,and provide user support.

- Customer evaluation-tasks required to obtain customer feedback based on evaluation of the software representations created during the engineering stage and implemented during the installation stage.

Advantages Of Spiral Model

Estimates (i.e budget, schedule, etc) become more realistic as work progresses, because more important issues are discovered earlier.

2. It is more able to cope with the changes that software development generally entails.

3. Software engineers can get their hands in and start working on a project earlier.

Disadvantages Of Spiral Model

1. Highly customized limiting re-usability.

2. Applied differently for each application.

3. Risk of not meeting,budget or schedule.