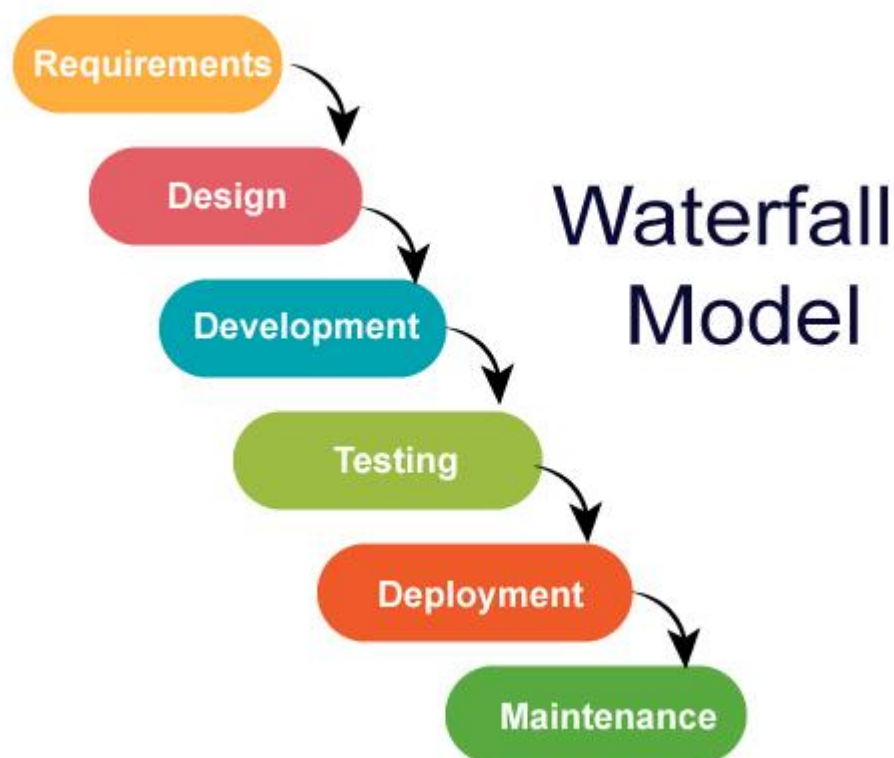


## **WATERFALL MODEL:**

The Waterfall Model was the first Process Model to be introduced. 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.

In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model the outcome of one phase acts as the input for the next phase sequentially.



### **Requirement Gathering:**

All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

### **System Design:**

The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

### **Implementation:**

With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

### **Integration and Testing:**

All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

**Deployment of system:**

Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

**Maintenance:**

There are some issues which come up in the client environment. To fix those issues, patches are release. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

## Waterfall Model – Advantages

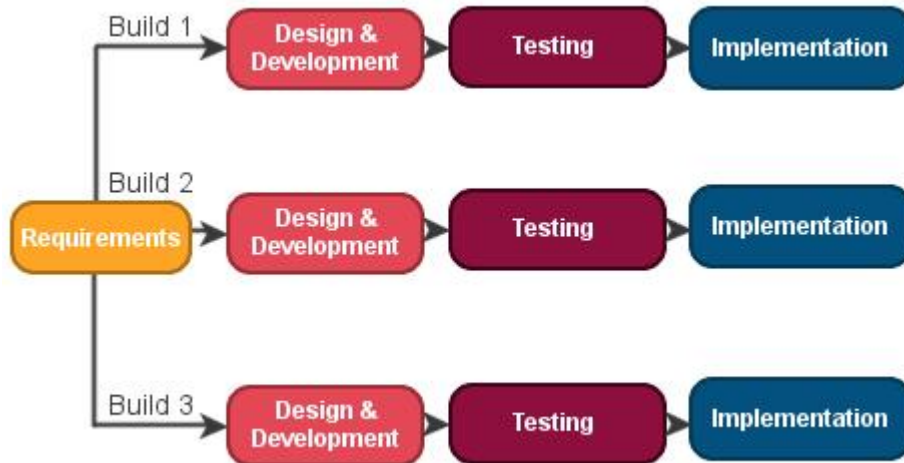
- 1.Simple and easy to understand and use
- 2.Each phase has specific deliverables and a review process.
- 3.Phases are processed and completed one at a time.
- 4.Works well for smaller projects where requirements are very well understood.
- 5.Clearly defined stages.
- 7.Easy to arrange tasks.
- 8.Process and results are well documented.
- 9.It gives easy to control and clarity for the customer due to a strict reporting system.

## Waterfall Model - Disadvantages

- 1.No working software is produced until late during the life cycle.
- 2.High amounts of risk and uncertainty.
- 3.Not a good model for complex and object-oriented projects.
- 4.Poor model for long and ongoing projects.
- 5.It is difficult to measure progress within stages.
- 6.Cannot accommodate changing requirements.
- 7.Adjusting scope during the life cycle can end a project.
- 8.Since the testing done at a later stage, it does not allow identifying the challenges and risks in the earlier phase, so the risk reduction strategy is difficult to prepare.

## ITERATIVE MODEL:

The iterative process model is the implementation of the software development lifecycle in which the initial development is started based on the initial requirement and more features are added to the base software product with the ongoing iteration until the final system is created.



**Requirement gathering & analysis:** In this phase, requirements are gathered from customers and check by an analyst whether requirements will fulfil or not. Analyst checks that need will achieve within budget or not. After all of this, the software team skips to the next phase.

**2. Design:** In the design phase, team design the software by the different diagrams like Data Flow diagram, activity diagram, class diagram, state transition diagram, etc.

**3. Implementation:** In the implementation, requirements are written in the coding language and transformed into computer programmes which are called Software.

**4. Testing:** After completing the coding phase, software testing starts using different test methods. There are many test methods, but the most common are white box, black box, and grey box test methods.

**5. Deployment:** After completing all the phases, software is deployed to its work environment.

**7. Maintenance:** In the maintenance phase, after deployment of the software in the working environment there may be some bugs, some errors or new updates are required. Maintenance involves debugging and new addition options.

## Advantage:

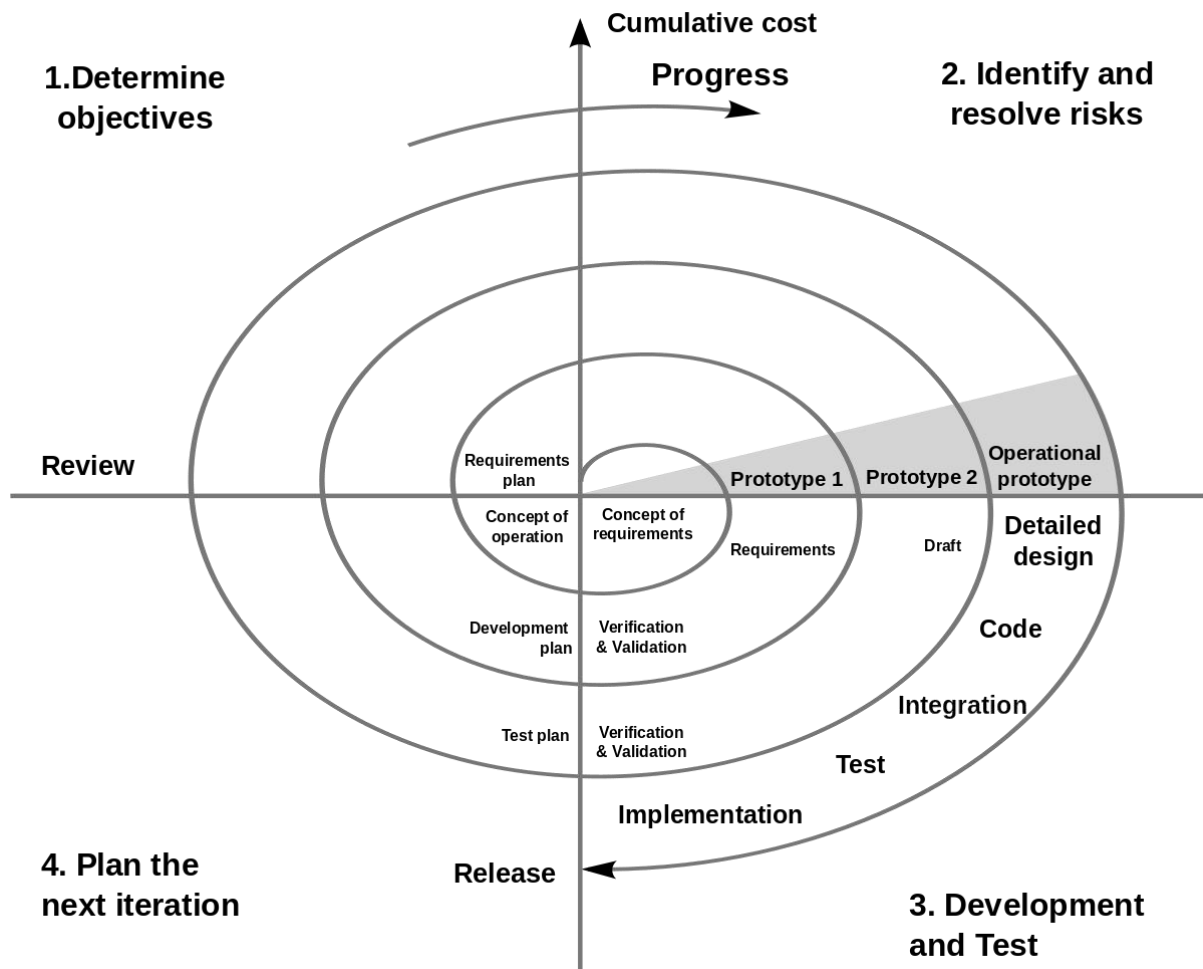
1. Testing and debugging during smaller iteration is easy.
2. A Parallel development can plan.
3. It is easily acceptable to ever-changing needs of the project.
4. Risks are identified and resolved during iteration.
5. Limited time spent on documentation and extra time on designing.

## Disadvantage:

1. It is not suitable for smaller projects.
2. More Resources may be required.
3. Design can be changed again and again because of imperfect requirements.
4. Requirement changes can cause over budget.
5. Project completion date not confirmed because of changing requirements.

## **SPIRAL MODEL:**

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral.



**Objective setting:** Each cycle in the spiral starts with the identification of purpose for that cycle, the various alternatives that are possible for achieving the targets, and the constraints that exists.

**Risk Assessment and reduction:** The next phase in the cycle is to calculate these various alternatives based on the goals and constraints. The focus of evaluation in this stage is located on the risk perception for the project.

**Development and validation:** The next phase is to develop strategies that resolve uncertainties and risks. This process may include activities such as benchmarking, simulation, and prototyping.

**Planning:** Finally, the next step is planned. The project is reviewed, and a choice made whether to continue with a further period of the spiral. If it is determined to keep, plans are drawn up for the next step of the project.

The development phase depends on the remaining risks. For example, if performance or user-interface risks are treated more essential than the program development

risks, the next phase may be an evolutionary development that includes developing a more detailed prototype for solving the risks.

The **risk-driven** feature of the spiral model allows it to accommodate any mixture of a specification-oriented, prototype-oriented, simulation-oriented, or another type of approach. An essential element of the model is that each period of the spiral is completed by a review that includes all the products developed during that cycle, including plans for the next cycle. The spiral model works for development as well as enhancement projects.

### Advantages:

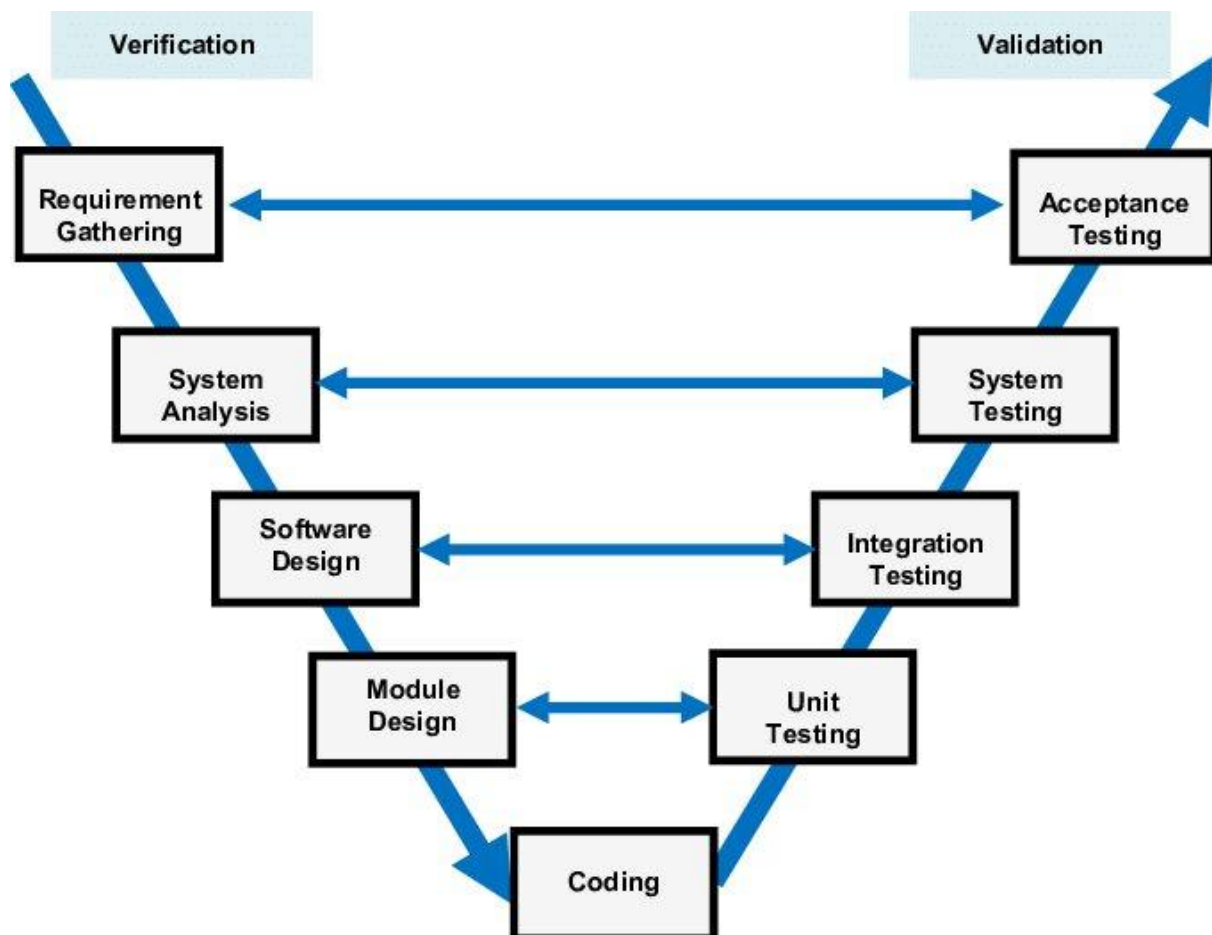
- High amount of risk analysis
- Useful for large and mission-critical projects.

### Disadvantages:

- Can be a costly model to use.
- Risk analysis needed highly particular expertise
- Doesn't work well for smaller projects.

### V MODEL:

V-Model also referred to as the Verification and Validation Model. In this, each phase of SDLC must complete before the next phase starts. It follows a sequential design process same as the waterfall model. Testing of the device is planned in parallel with a corresponding stage of development.



### Advantage:

1. Easy to Understand.
2. Testing Methods like planning, test designing happens well before coding.
3. This saves a lot of time. Hence a higher chance of success over the waterfall model.
4. Avoids the downward flow of the defects.
5. Works well for small plans where requirements are easily understood.

### Disadvantage:

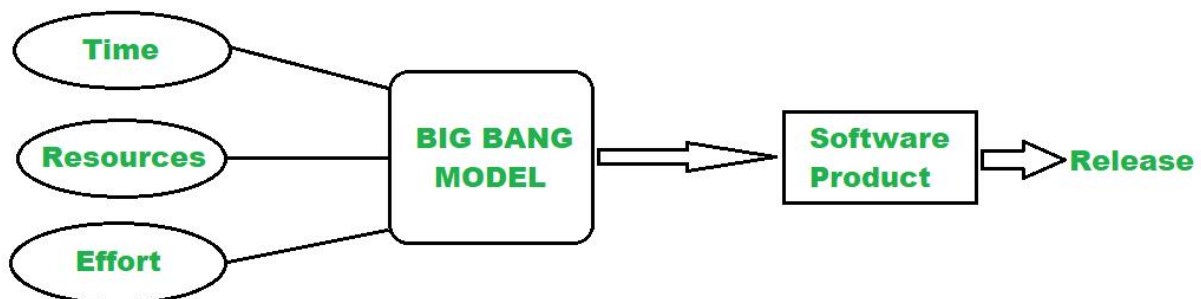
1. Very rigid and least flexible.
2. Not a good for a complex project.
3. Software is developed during the implementation stage, so no early prototypes of the software are produced.

4. If any changes happen in the midway, then the test documents along with the required documents, has to be updated.

### **BIGBANG MODEL:**

The Big Bang model is an SDLC model where we do not follow any specific process. The development just starts with the required money and efforts as the input, and the output is the software developed which may or may not be as per customer requirement. This Big Bang Model does not follow a process/procedure and there is a very little planning required. Even the customer is not sure about what exactly he wants and the requirements are implemented on the fly without much analysis.

This model is ideal for small projects like academic projects or practical projects. One or two developers can work together on this model.



### **Advantage:**

1. There is no planning required.
2. Simple Model.
3. Few resources required.
4. Easy to manage.
5. Flexible for developers.
6. It is a good learning aid for new comers or students.



## Disadvantage:

1. There are high risk and uncertainty.
2. Not acceptable for a large project.
3. Not a good model for complex and object-oriented projects.
4. If requirements are not clear that can cause very expensive.