# Software development life cycle Models


Hundreds of different kinds of models are known and used. But many are minor variations on a smaller number of basic models. We will concentrate on the basic models in detail.

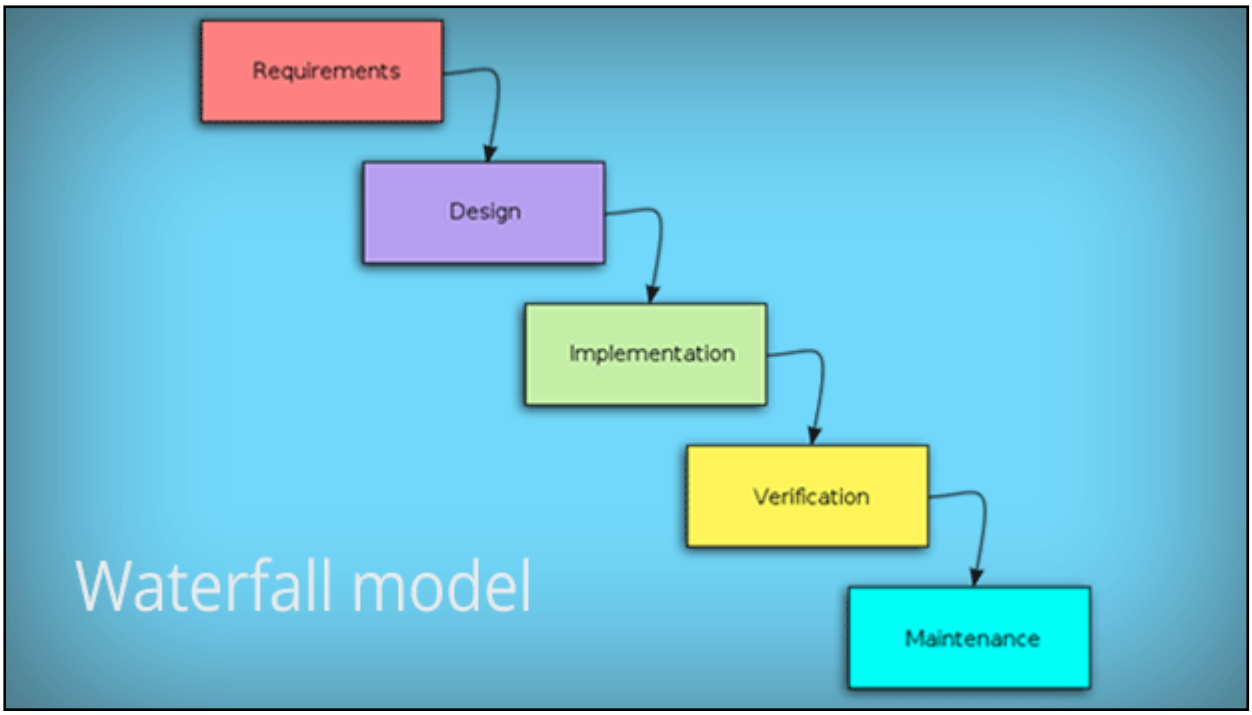
## What is a life cycle model?

A life-cycle model is a sequence of activities carried out in a software project, and the order of these activities.

Project Plan = life cycle model + Project parameters

## The Waterfall Model:

A classic life cycle model which is widely used where, a project starts with an initial phase, upon completion of the phase, it moves to the next phase and so on to the third phase.




### Advantages:

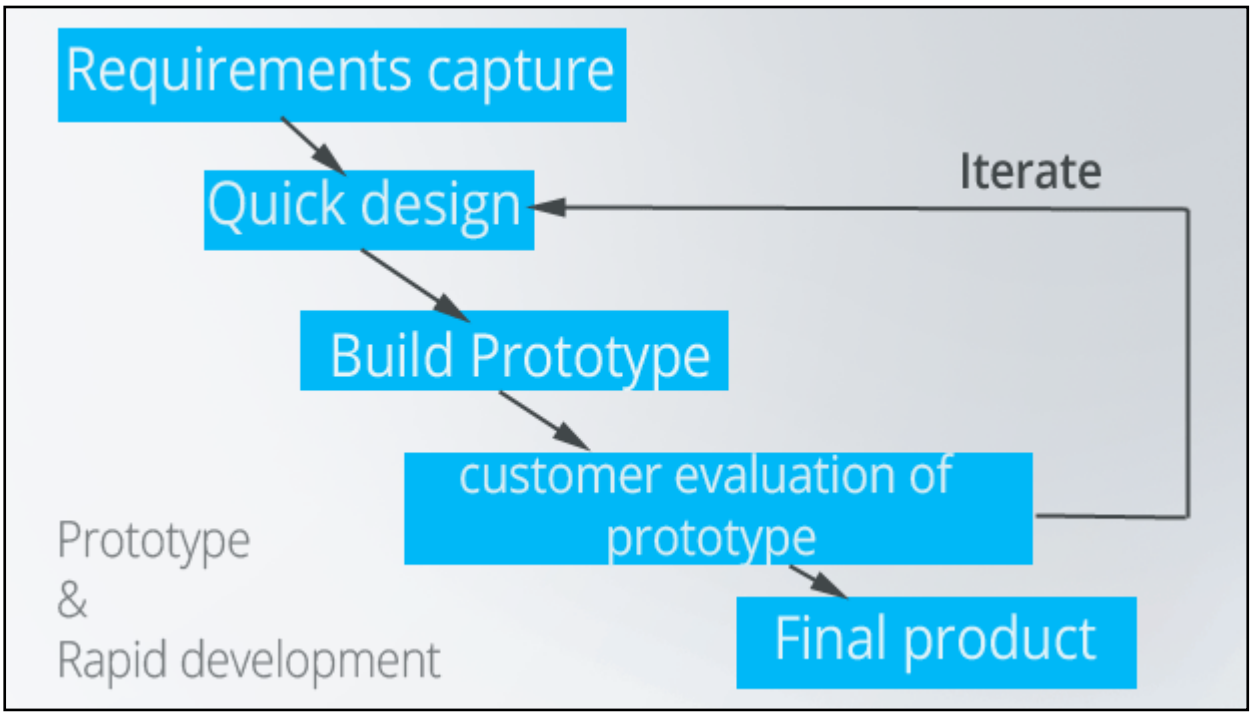
- Easy to understand and implement.
- Has good habits – requirements-before-design, design-before-coding.
- Project monitoring and maintenance is easy.
- In this model phases are processed and completed one at a time. No overlapping of phases.
- Waterfall model works well for smaller projects where requirements are clear.

### Disadvantages:

- When the system is under testing phase, it is very difficult to change something that was not in requirements stage.
- No working software would be available until the last phase.
- High risk involved.
- Not suitable for the projects where requirements are supposed to be changed.

## Prototype and RAD models:

In this type, Customers are non-technical and usually don't know what they want. Hence, It requires frequent interactions with the customer while gathering requirements and a prototype is produced. This prototype is used to derive SRS docs.




**Advantages:**

- Reduces risk of incorrect user requirements.
- Good where requirements are changing.

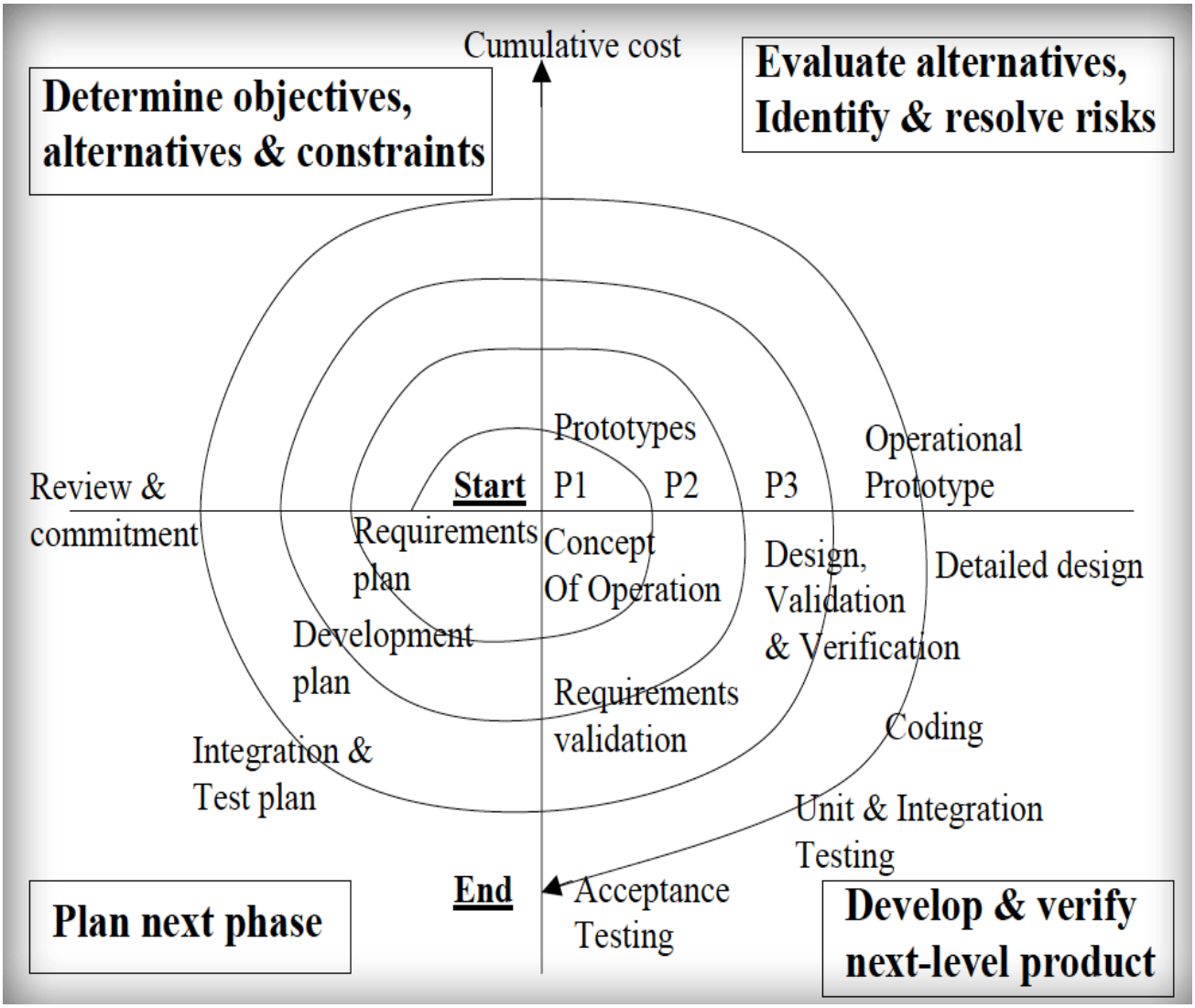
**Disadvantages:**

- An unstable or badly implemented prototype often becomes the final product
- Difficult to know about the project duration.
- Easy to fall back into code-and-fix without proper requirements analysis, design, customer evaluation and feedback.

# Spiral Model:

In this model ,End-user requirements are hard to obtain/define. So, its good to develop a software in experimental way: e.g,

1. Build some software
2. See if it meets customer requirements
3. If no goto step 1, else stop.




In 1988 Boehm developed the spiral model which includes risk analysis and risk management.

**Advantages:**

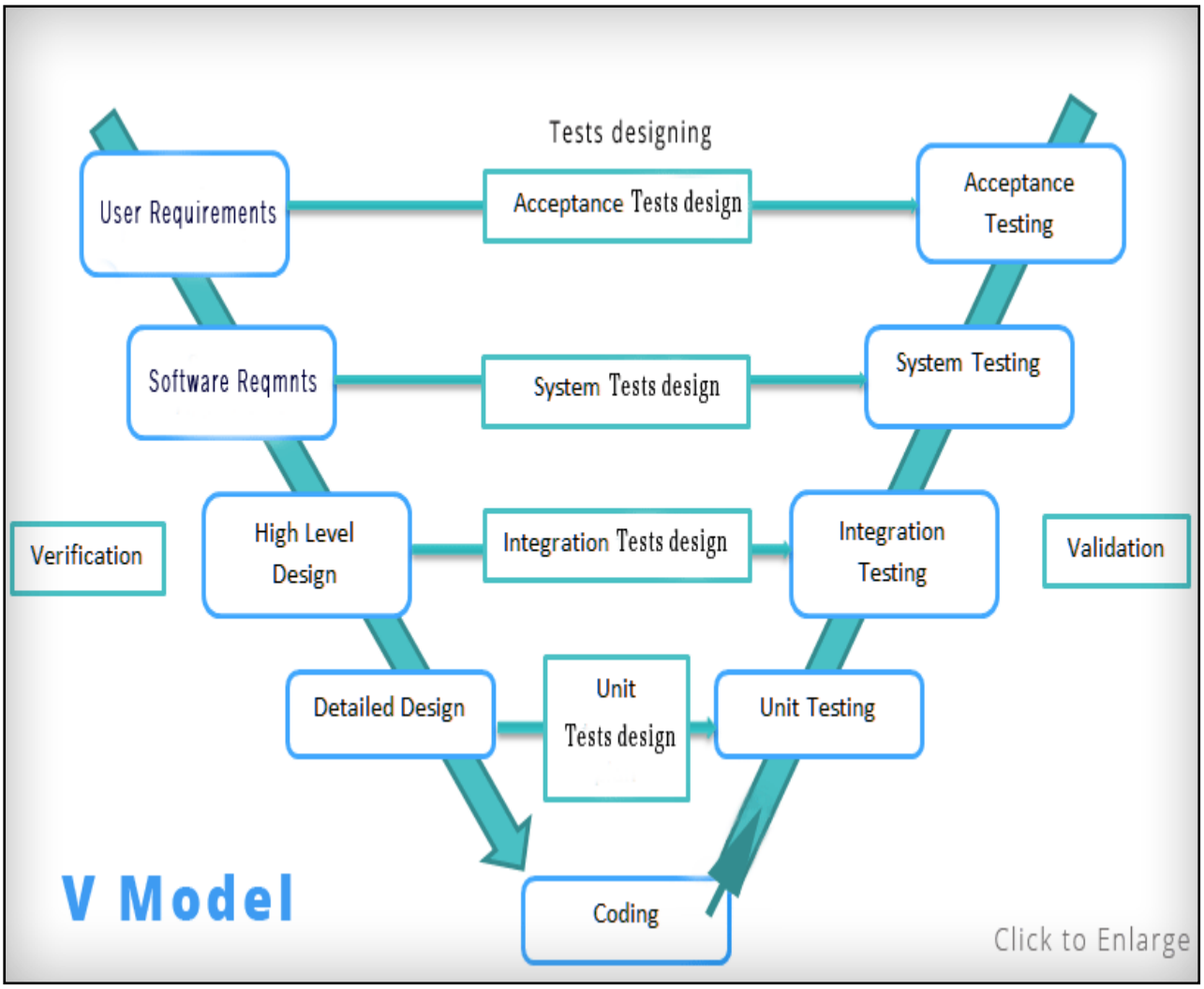
1. Realism: the model accurately reflects the iterative nature of software development on projects with unclear requirements.
2. Flexible: incorporates the advantages of the waterfall and RAD, prototypes.
3. Comprehensive model decreases risk.
4. Good project visibility.

### Disadvantages:

1. Needs technical expertise in risk analysis to really work
2. Model is poorly understood by nontechnical management, hence not so widely used
3. Complicated model needs competent professional management.

# V Model:

In water fall model, testing is a post development activity. The other models went a step further and tried to break up the product into increments. However, V model brings in different types of testing applicable at different levels for productive output.




1. The V model splits testing into two parts –

- **Actual activities + design**
- **Execution of tests.**

**Test design in this model is done early and test execution at the later stage.**

**Advantages:**

- Simple and easy to use.
- Testing activities like planning, test designing occur before coding. Which overcomes the negatives of waterfall model.

- Defects are found at early stage.
- Works good for projects where, requirements are clear

## Disadvantages:

- Rigid and least flexible.
- Software is developed during the implementation phase, so no early prototypes of the software are produced.
- If changes occur in the middle, then the test documents along with requirement documents are to be updated.