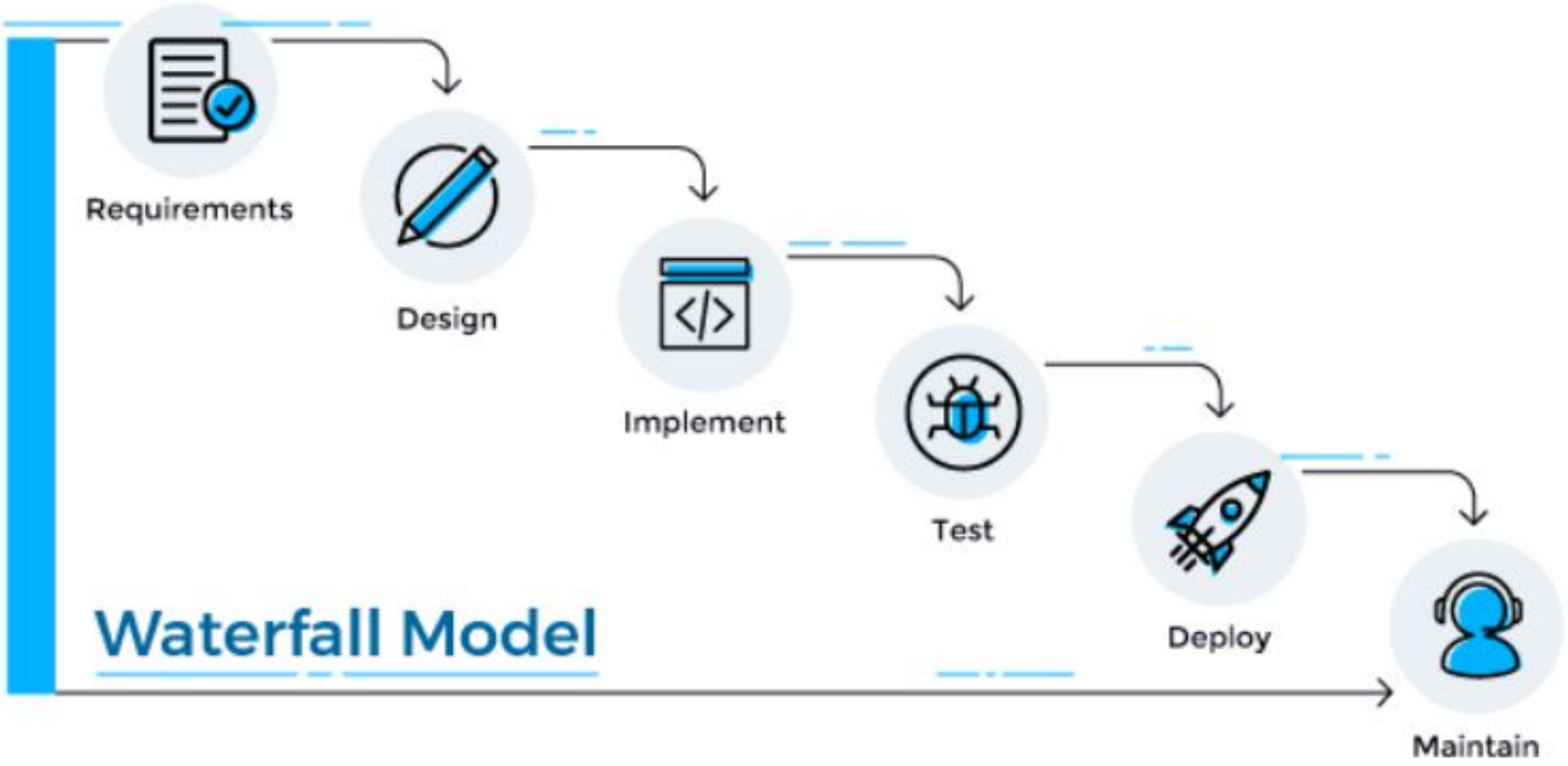
SDLC MODELS

- There are various software development life cycle models defined and designed which are followed during the software development process.

- These models are also referred as Software Development Process Models". Each process model follows a Series of steps unique to its type to ensure success in the process of software development.

Waterfall Model

Waterfall Model



- In "The Waterfall" approach, the whole process of software development is divided into separate phases.

- In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

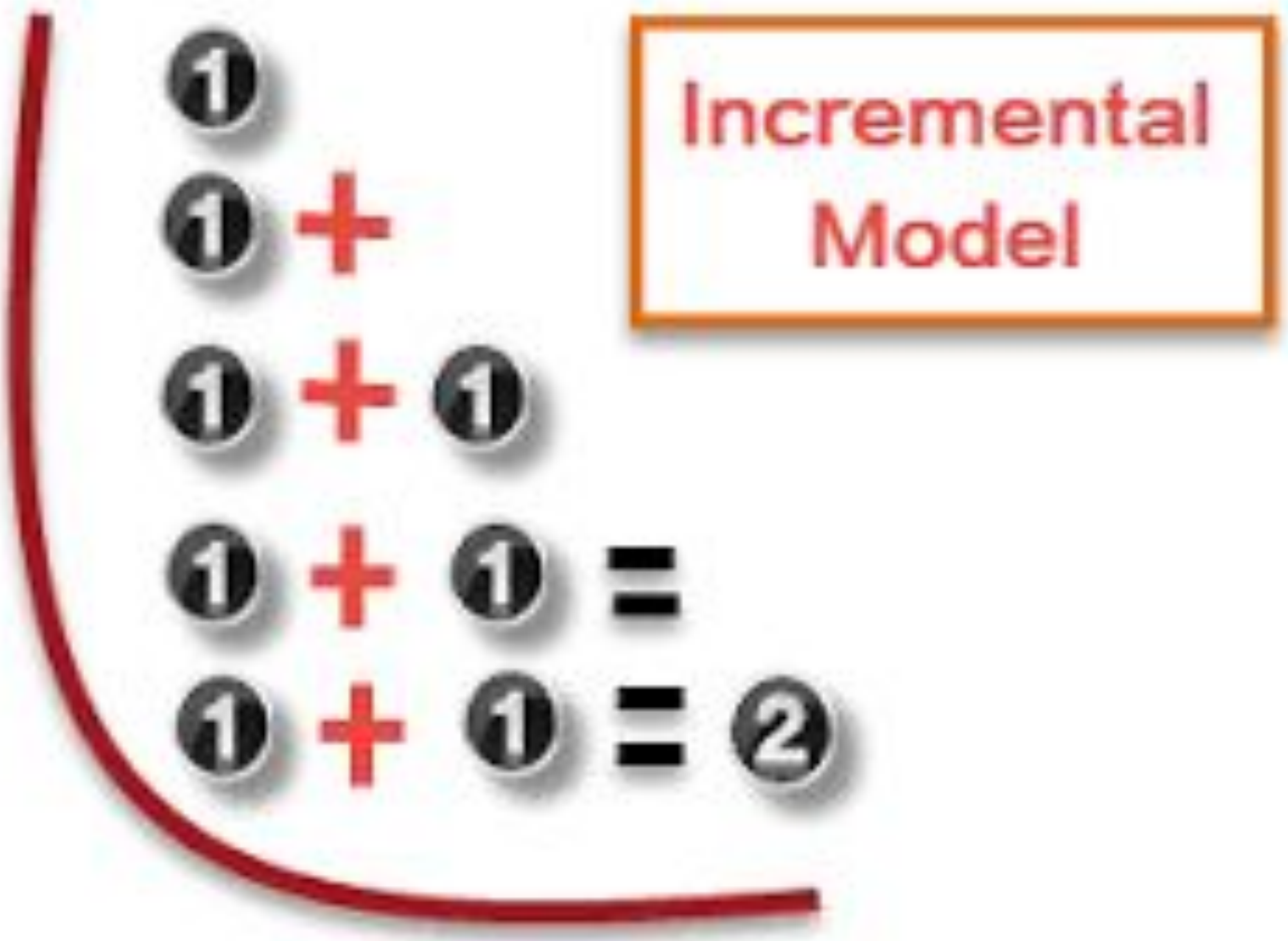
- The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

Waterfall Model - Application

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

- Waterfall model – sequential model

  output of one phase is given as the input of another phase


Incremental Model
1
1 +
1 + 1
1 + 1 =
1 + 1 = 2

Incremental model

- One of the SDLC model
- Multiple releases are there
- Used for long term project
- Used if the requirements are not clear
- Can accommodate changes
- Customer is considered

www.t4tutorials.com
Increment 1
Increment 2
Increment 3
Increment 4

Advantages

- Risk handling
- Flexibility in requirement

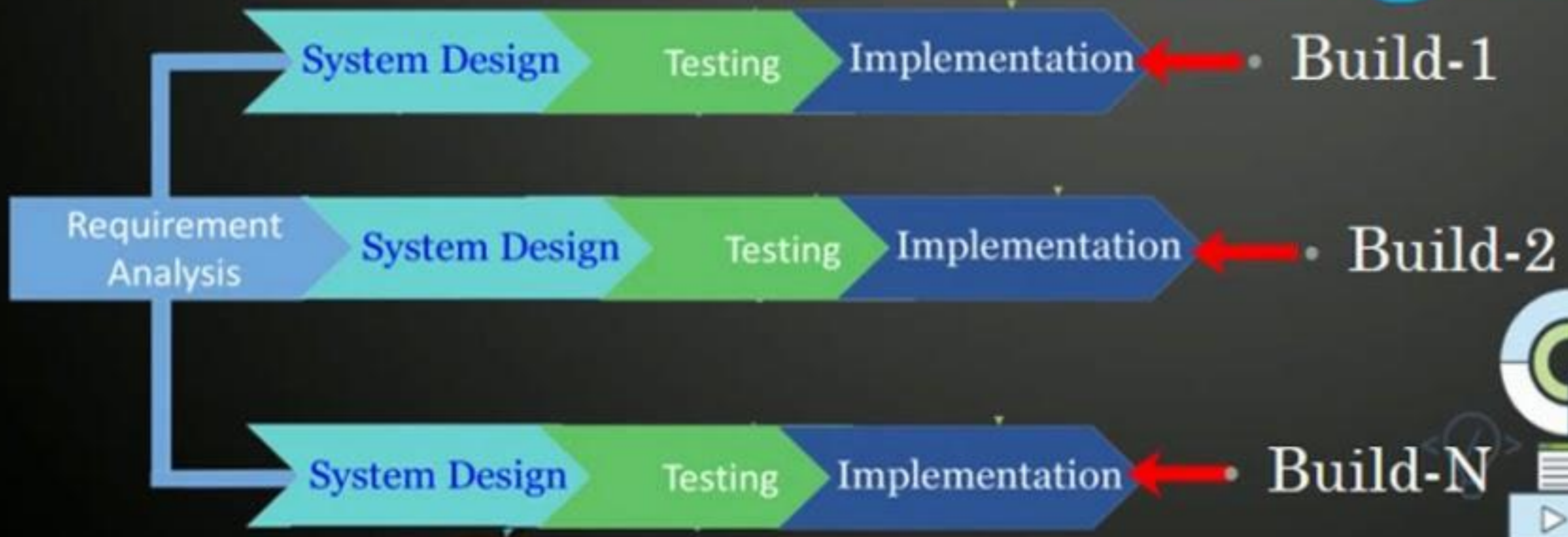
Disadvantage

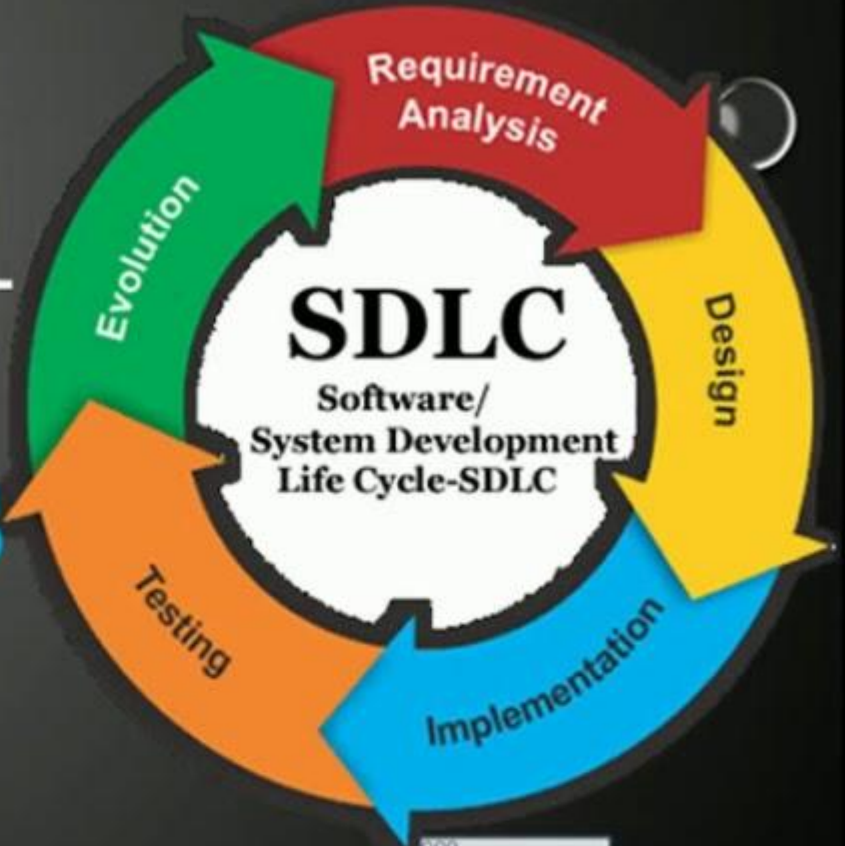
- Complex
- Expensive so not suitable for small projects
- Difficulty in time management

Iterative Model

Software/System Development Life Cycle

Phases/Stages





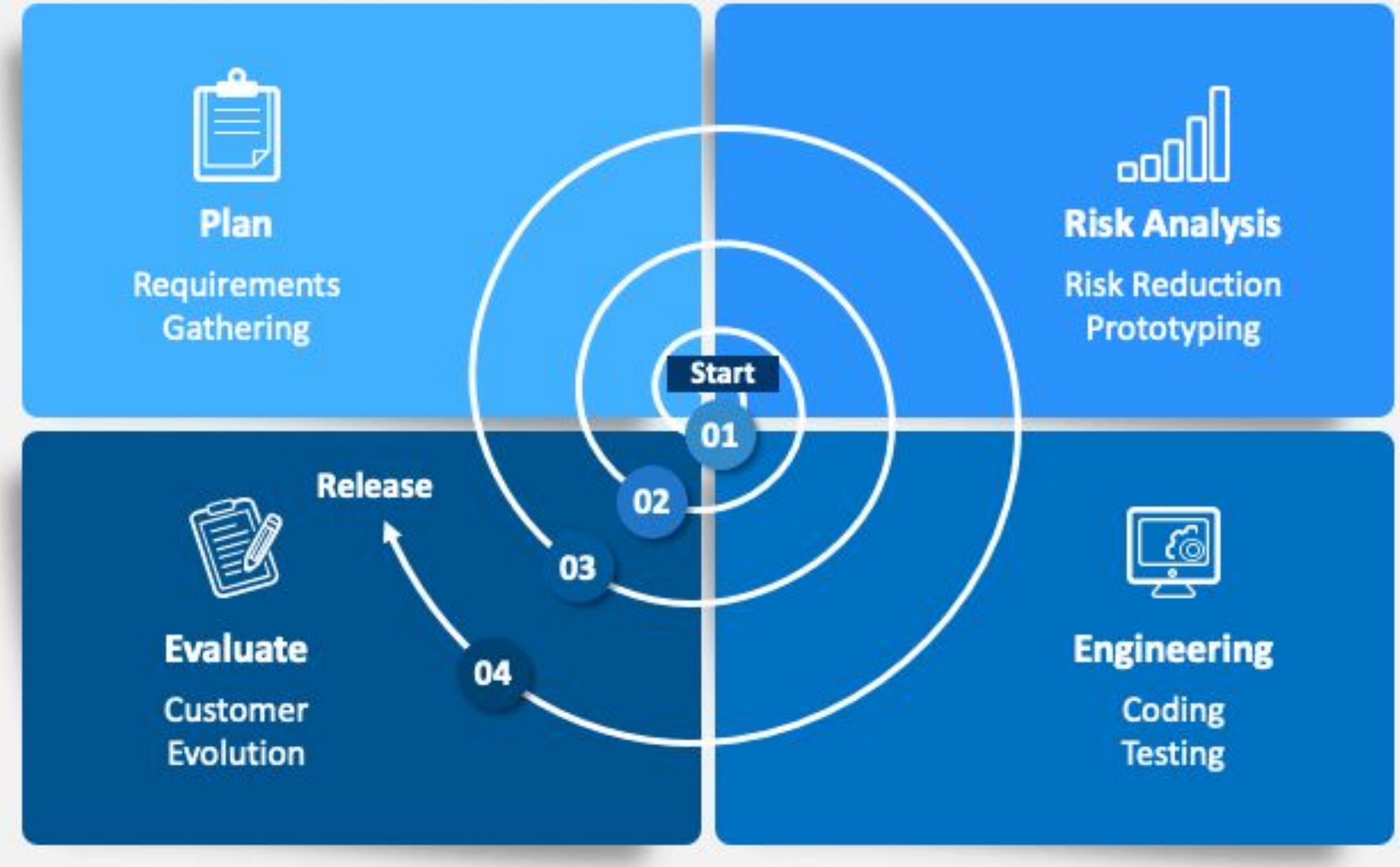
Iterative model/ Prototype Model

- One of the SDLC model
- Multiple releases are there
- Used if the requirements are not clear
- A working prototype is released initially
- Used for long term projects

- The most significant disadvantage of previous models (waterfall and spiral) is that there were **lots of customer rejection** that happens after the application was developed, and there was no involvement of the customers in between the project.
- Hence, they started the new approach, which is known as the **prototype model**. In this, we will collect the requirements from the customer and prepare a **prototype (sample)**, and get it reviewed and approved by the customer. And only when they satisfied, we will start working on the original projects so that there won't be any customer rejection.

SPIRAL MODEL IN SDLC

Enter your sub headline here



Plan

Requirements Gathering

Risk Analysis

Risk Reduction Prototyping

Evaluate

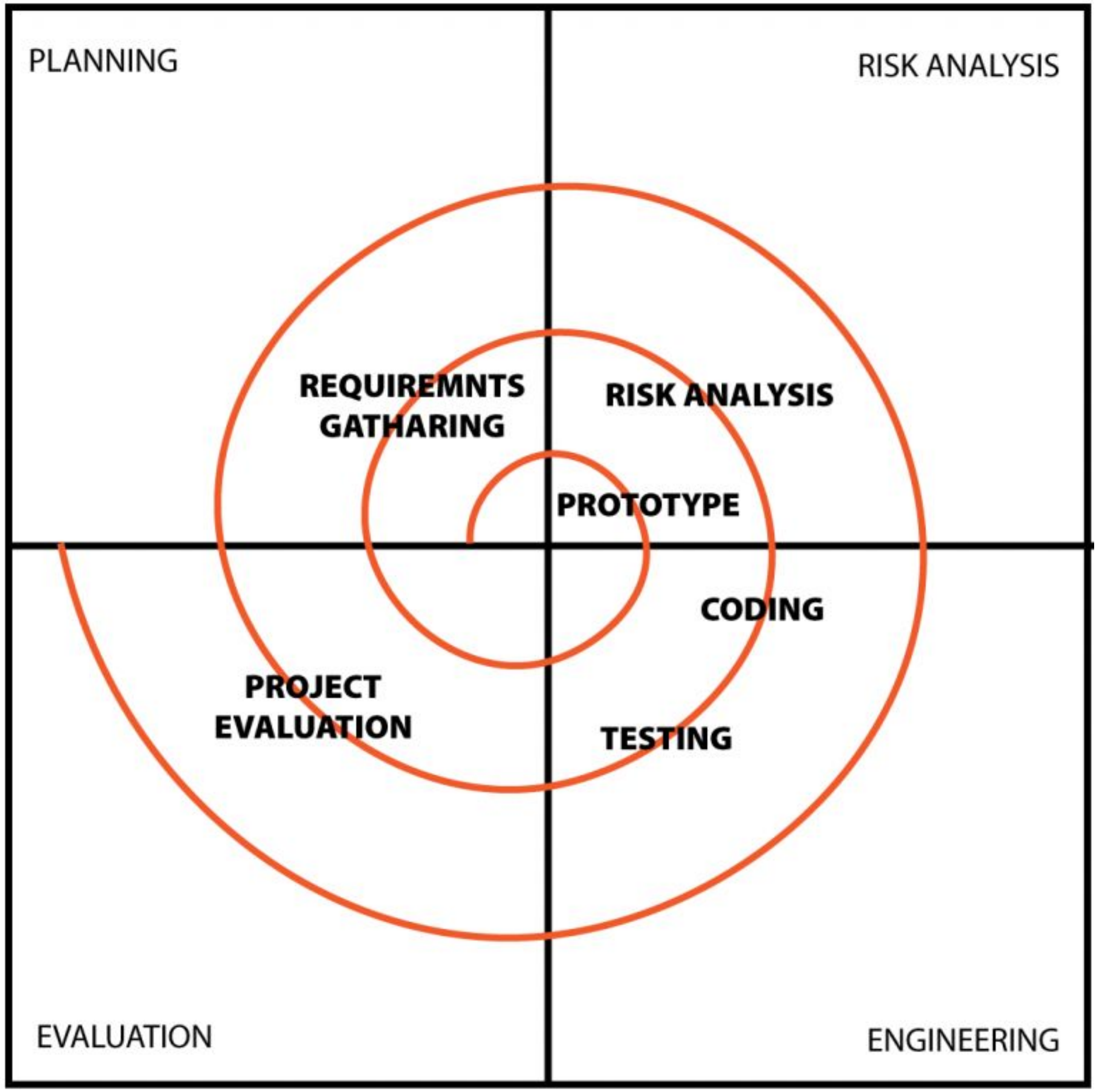
Customer Evolution

Engineering

Coding Testing

Spiral Model

1. Planning
2. Risk Analysis
3. Engineering
4. Evaluation



- Suitable for complex project

- Requirements are changing from the customer side

- Suitable for the projects with high chance of risk
- One loop is one iteration

- Planning

  Gathering requirements

  Analysing the requirements

  Planning based on the requirements
- Risk

  Analysing the risk

  Handling the risk
- Engineering

  Designing and coding

  Architectural design and module logic design are done

  After some iteration final design is made

  Last of this quadrant prototype is developed

  Coding Start

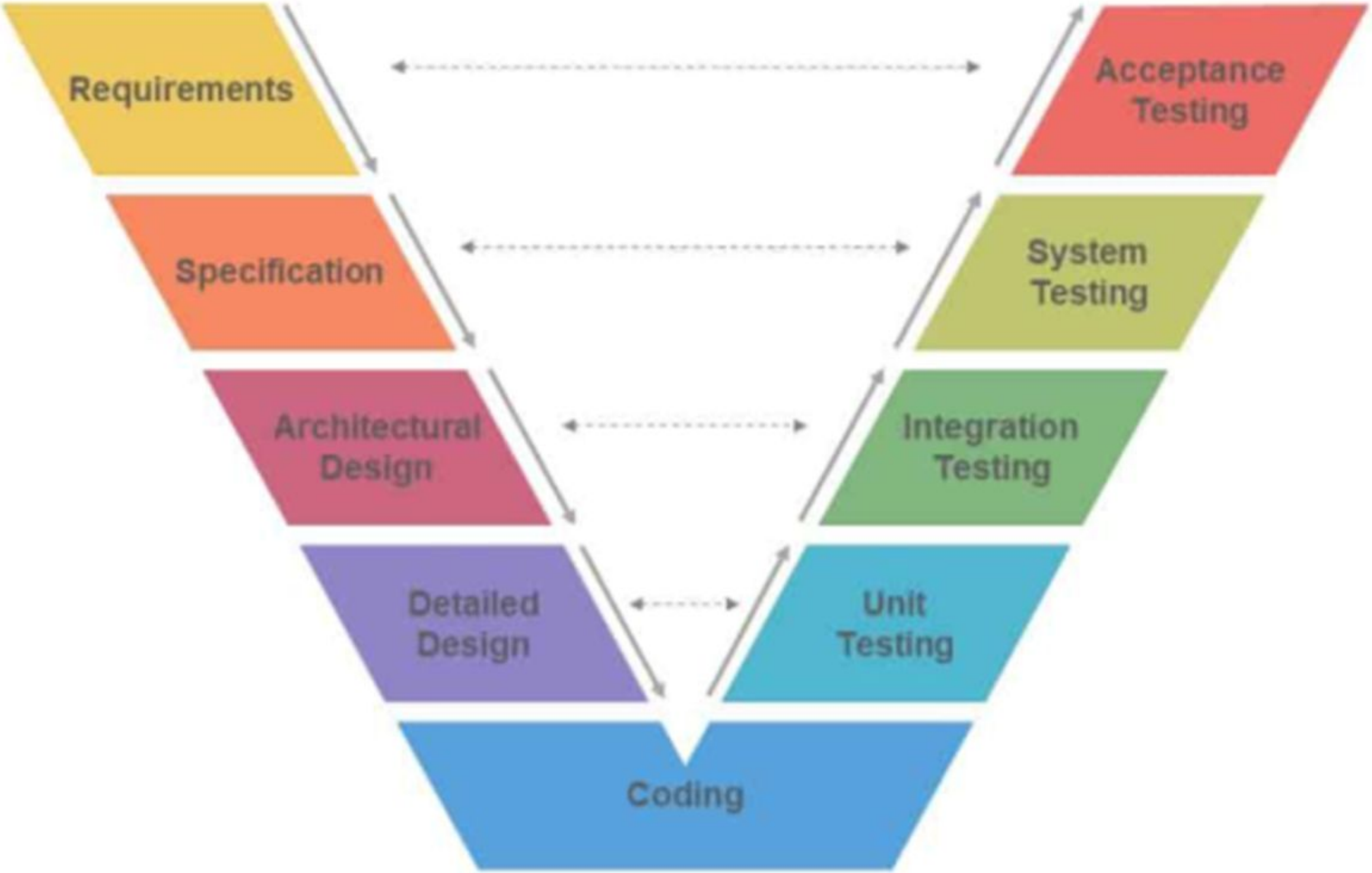
  At last Software or application is developed completely

  Testing is done
- Evaluation

  Customer uses the application, feed back from customer is collected

Diagram

V-MODEL SOFTWARE DEVELOPMENT



V model/ Verification & validation/ QA & QC Model

- One of the SDLC Model
- Testing activities are done parallel
- This model can be used for short term projects
- This model can be used if the requirements are clear

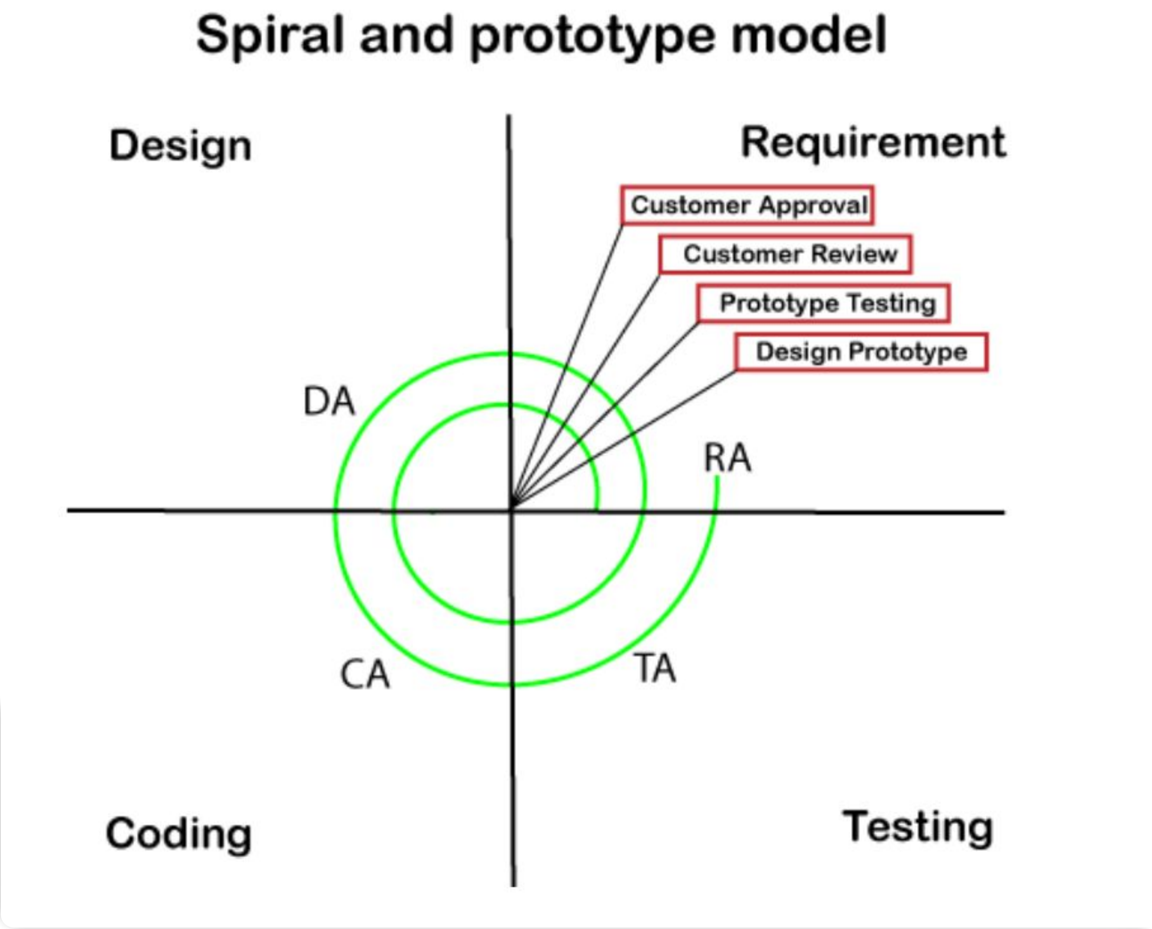
Hybrid Model

- The hybrid model is the combination of two or more primary (traditional) models and modifies them as per the business requirements.
- The hybrid model is mainly used for small, medium, and large projects. It focuses on the risk management of the product.
- The most commonly used combination of two models is as follows:
- **Spiral and prototype**
- **V & V and Prototype**

Spiral & Prototype

- The spiral and prototype model is used for the below conditions:
- When the customer gives requirements in stages, and we develop the product in stages using this model.
- When the customer is very new to the software industry and not clear about the requirements.
- When the developers are new to particular software.

Process of Spiral and Prototype model

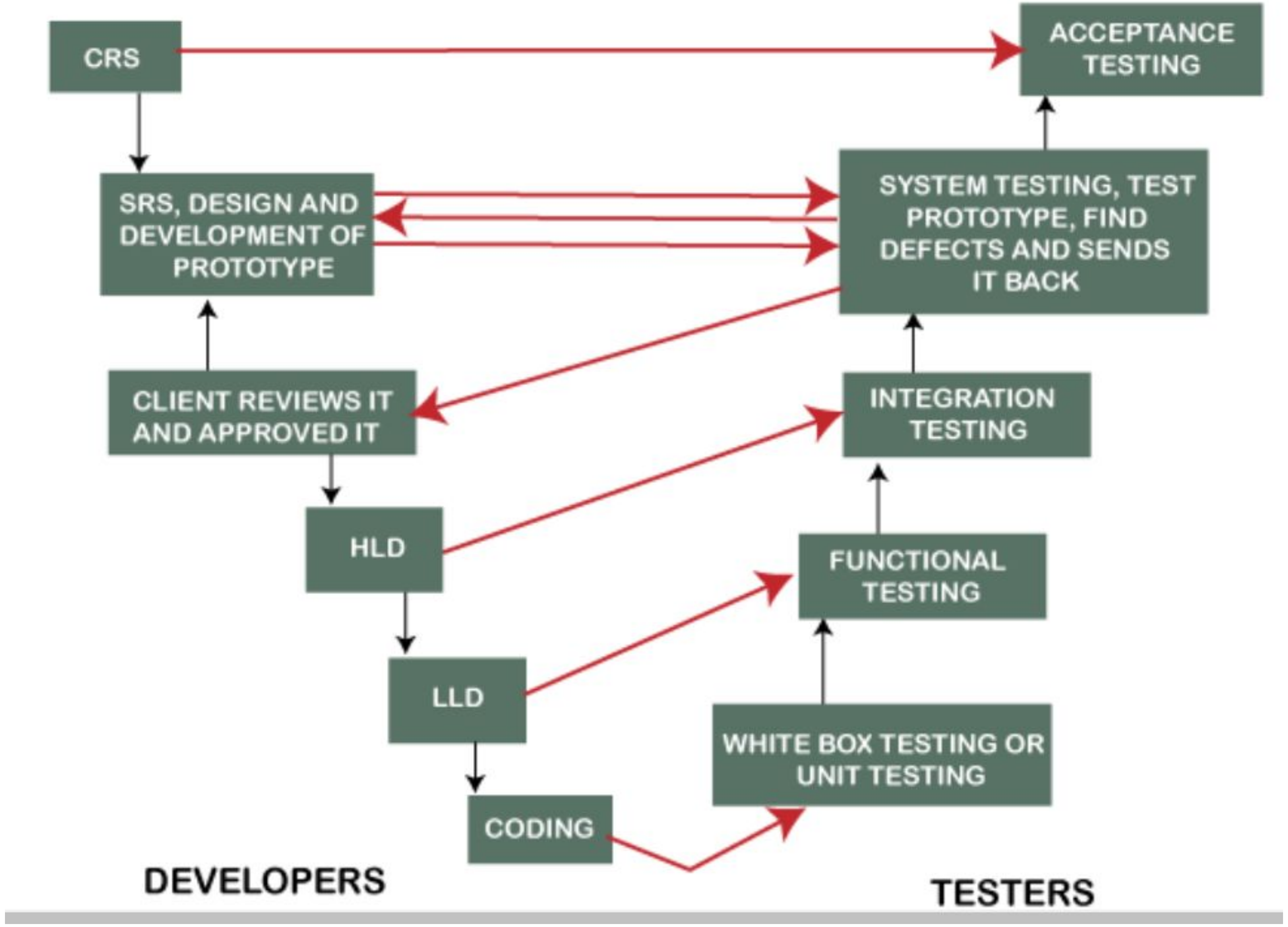


- The process of this model starts with **collecting the requirements** from the customer for the different modules like A, B, and C
- After collecting the business needs of the software, we will **create the prototype A**.
- Once we develop the Prototype, we will **test the Prototype A**.
- After successfully testing the Prototype, we will send it to the customer for their **review and approval**.
- Once they reviewed and approved the Prototype, we will design that Prototype for the actual module.
- Once the designing phase completes, the **developer starts writing the code** for the modules.
- After the completion of development, it will send it to the testing team, where **they will test the module**.
- And when the testing phase is done, it will **deploy to the customer**.

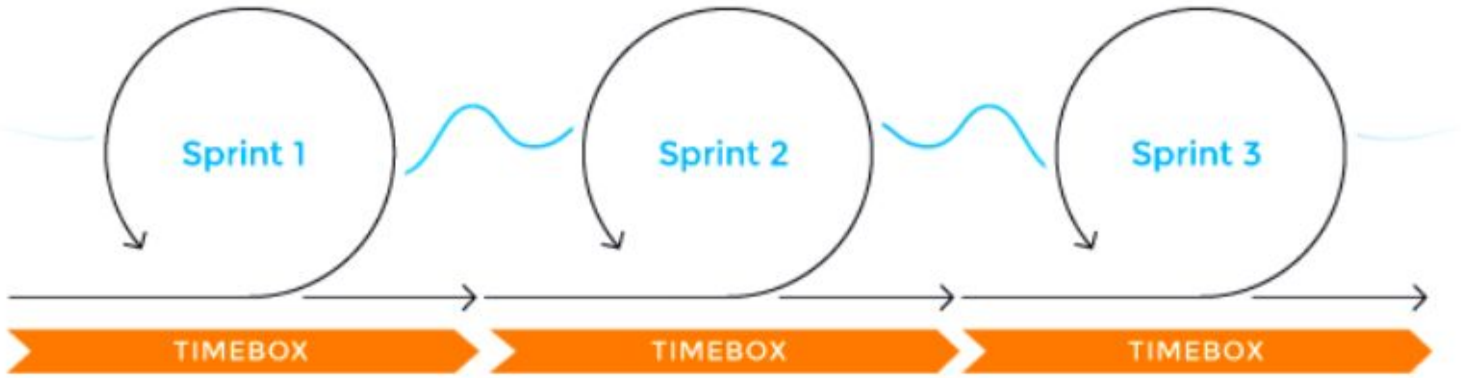
V & V and prototype Model

- We go for this model for the following reasons:
- When the customer and developers are both new to the industry.
- When the clients are expecting a very high-quality product within the required time because every phase is tested, and the developer and testing team are working parallelly.

V & V and prototype model


CRS
ACCEPTANCE TESTING
SRS, DESIGN AND DEVELOPMENT OF PROTOTYPE
SYSTEM TESTING, TEST PROTOTYPE, FIND DEFECTS AND SENDS IT BACK
CLIENT REVIEWS IT AND APPROVED IT
INTEGRATION TESTING
HLD
FUNCTIONAL TESTING
LLD
WHITE BOX TESTING OR UNIT TESTING
CODING
DEVELOPERS
TESTERS

Agile Methodology



Agile Model

- One of the SDLC Model
- Used for long term project
- Used if the requirements are not clear
- Client interaction is more
- Client will get the application for testing in the first release itself

Agile Methodology/ Type/ Framework

5 types

- Scrum
- Extreme Programming
- Adaptive Software Development (ASD)
- Dynamic Software Development Method (DSDM)
- Kanban

Agile Terms

- User Story
  - Is the functionality of the application
  - Similar to a feature
- Sprint
  - One development cycle in Scrum
  - Include requirement, designing, coding, testing and client testing
  - Duration – 2 to 4 weeks
- Story Point
  - It is the complexity of each user stories

Different Roles

- Product Owner (PO)
  - Client
- Scrum Master (SM)
  - He should manage the team
  - Should sort out the problems
- Team
  - 8 to 12 members

Different Meetings

- Planning Meeting
  - Product owner will explain user stories with the team
  - Team will identify the user stories for the particular sprint
- Daily Stand Up Meeting
  - Team will discuss the following
  - What all things they did yesterday
  - What they will do today
  - What all issues they are facing
- Review Meeting
  - Team will show the completed user stories with the product owner.
  - Product owner will test the completed user stories
- Retrospective Meeting
  - Team will discuss the following
  - What all things went good?
  - What all things went bad
  - How to prevent it in feature

- Product Backlog
  - The pending user stories in the product

- Sprint Backlog
  - The pending user stories in the sprint

- Team Velocity
  - How much effort a team can handle in one sprint.

  (Based on the days and number of team members involved)

- Burn down chart/ Story Board
  - It shows the progress of sprint