



LEARN WITH RABBIL

# Foundation Software Engineering



# WHAT IS SOFTWARE ENGINEERING

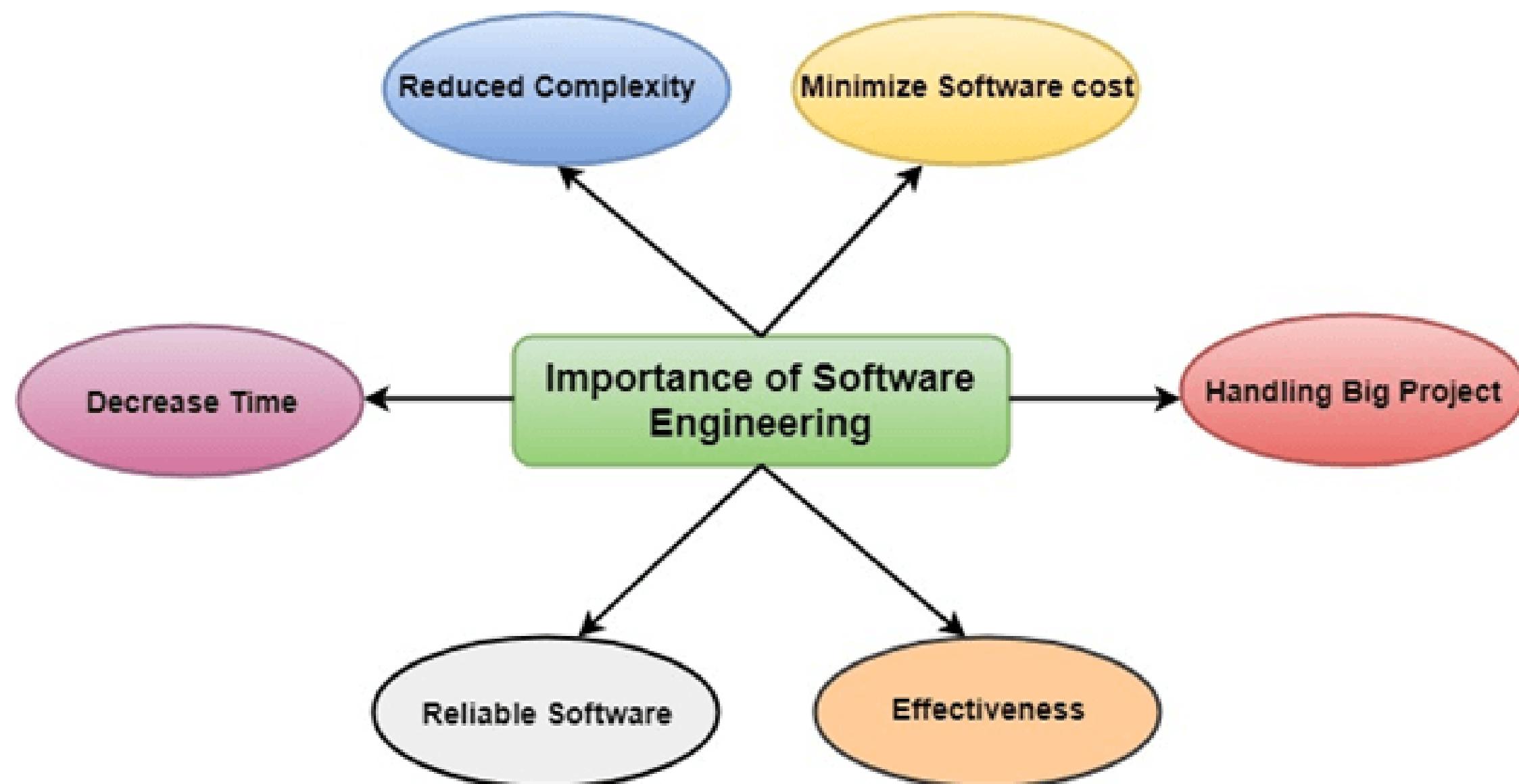
The **software** is a collection of integrated programs.

**Engineering** is the application of scientific and practical knowledge to invent, design, build, maintain, and improve frameworks, processes,





# Importance of Software Engineering





# Requirement Engineering

**01**

## Feasibility Study

The objective behind the feasibility study is to create the reasons for developing the software that is acceptable to users, flexible to change and conformable to established standards.

**03**

## Software Requirement Specification

Software requirement specification is a kind of document which is created by a software analyst after the requirements collected from the various sources - the requirement received by the customer written in ordinary language.

**05**

## Software Requirement Management

Requirement management is the process of managing changing requirements during the requirements engineering process and system development

**02**

## Requirement Elicitation and Analysis

This is also known as the gathering of requirements. Here, requirements are identified with the help of customers and existing systems processes, if available

**04**

## Software Requirement Validation

After requirement specifications developed, the requirements discussed in this document are validated. The user might demand illegal, impossible solution or experts may misinterpret the needs.



LEARN WITH RABBIL

# Requirement Engineering

## Feasibility Study

### Technical Feasibility

Technical feasibility evaluates the current technologies, which are needed to accomplish customer requirements within the time and budget

### Operational Feasibility

Operational feasibility assesses the range in which the required software performs a series of levels to solve business problems and customer requirements.

### Economic Feasibility

Economic feasibility decides whether the necessary software can generate financial profits for an organization.





# Requirement Engineering

## Requirement Elicitation and Analysis

This is also known as the gathering of requirements. Here, requirements are identified with the help of customers and existing systems processes, if available.

## Problems of Elicitation and Analysis

- Getting all, and only, the right people involved.
- Stakeholders often don't know what they want
- Stakeholders express requirements in their terms.
- Stakeholders may have conflicting requirements.
- Requirement change during the analysis process.
- Organizational and political factors may influence system requirements.





# Requirement Engineering

## Software Requirement Specification

Software requirement specification is a kind of document which is created by a software analyst after the requirements collected from the various sources

## Data Flow Diagrams

Data Flow Diagrams (DFDs) are used widely for modeling the requirements. DFD shows the flow of data through a system. The system may be a company, an organization, a set of procedures, a computer hardware system, a software system, or any combination of the preceding. The DFD is also known as a data flow graph or bubble chart.

## Data Dictionaries

Data Dictionaries are simply repositories to store information about all data items defined in DFDs. At the requirements stage, the data dictionary should at least define customer data items, to ensure that the customer and developers use the same definition and terminologies.

## Entity-Relationship Diagrams

Another tool for requirement specification is the entity-relationship diagram, often called an "E-R diagram." It is a detailed logical representation of the data for the organization and uses three main constructs i.e. data entities, relationships, and their associated attributes.





# Requirement Engineering

## Software Requirement Validation

After requirement specifications developed, the requirements discussed in this document are validated. The user might demand illegal, impossible solution or experts may misinterpret the needs. Requirements can be the check against the following conditions -

- If they can practically implement
- If they are correct and as per the functionality and specially of software
- If there are any ambiguities
- If they are full
- If they can describe





# Requirement Engineering

## Software Requirement Management

Requirement management is the process of managing changing requirements during the requirements engineering process and system development.

- New requirements emerge during the process as business needs a change, and a better understanding of the system is developed.
- The priority of requirements from different viewpoints changes during development process.
- The business and technical environment of the system changes during the development.





# Software Development Life Cycle

A life cycle model represents all the methods required to make a software product transit through its life cycle stages.

**Stage1:** Planning and requirement analysis

**Stage2:** Defining Requirements

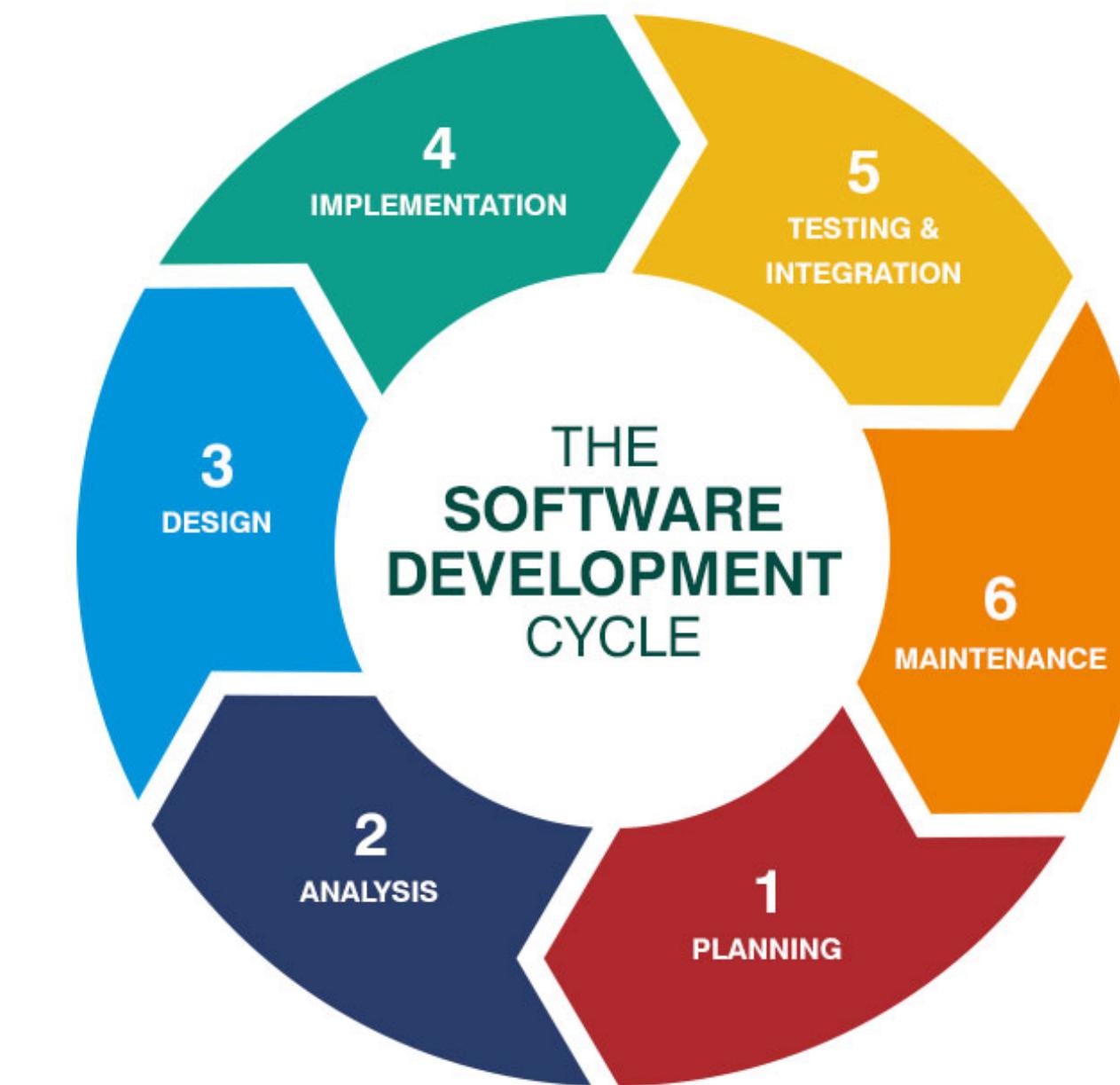
**Stage3:** Designing the Software

**Stage4:** Developing the project

**Stage5:** Testing

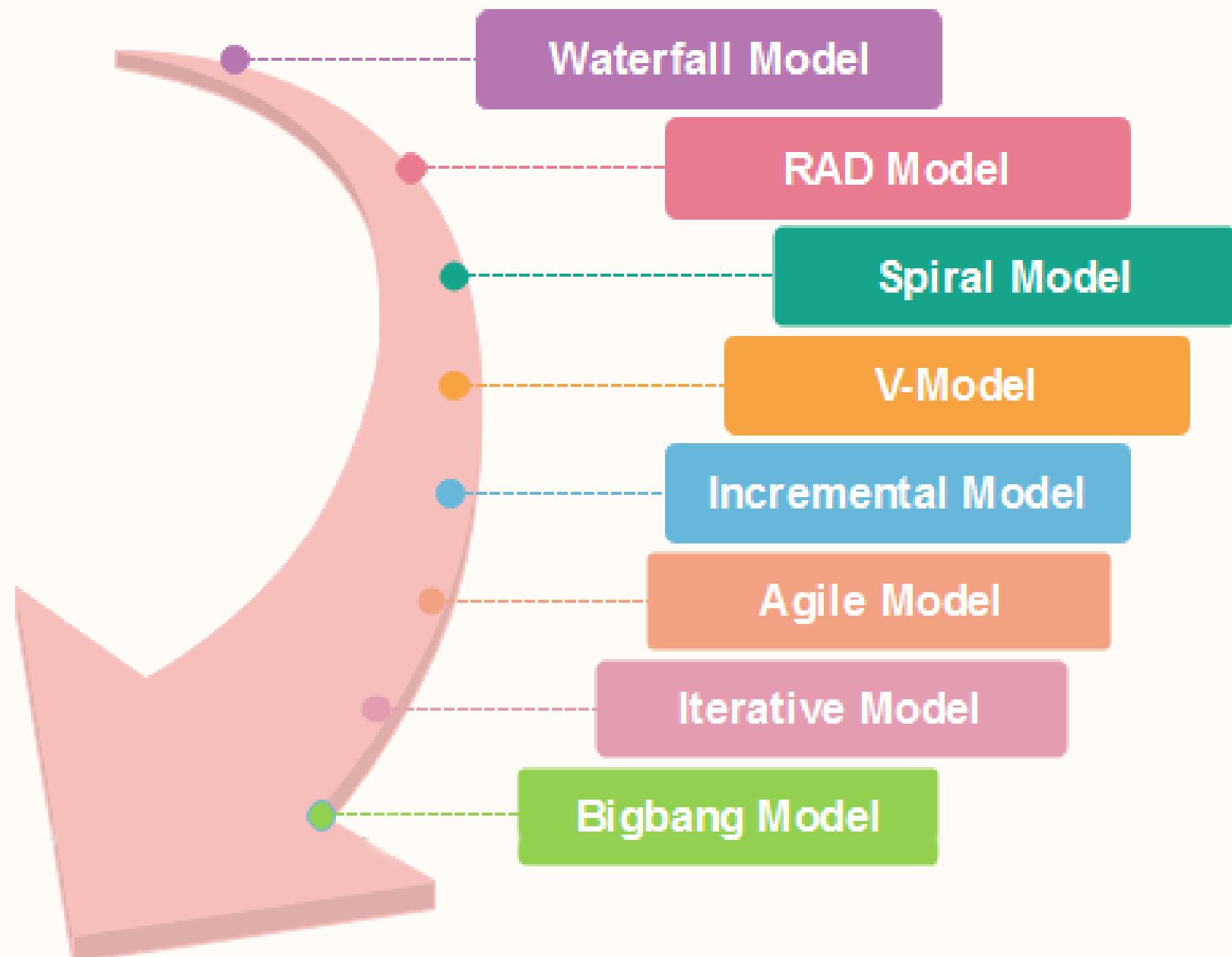
**Stage6:** Deployment

**Stage7:** Maintenance



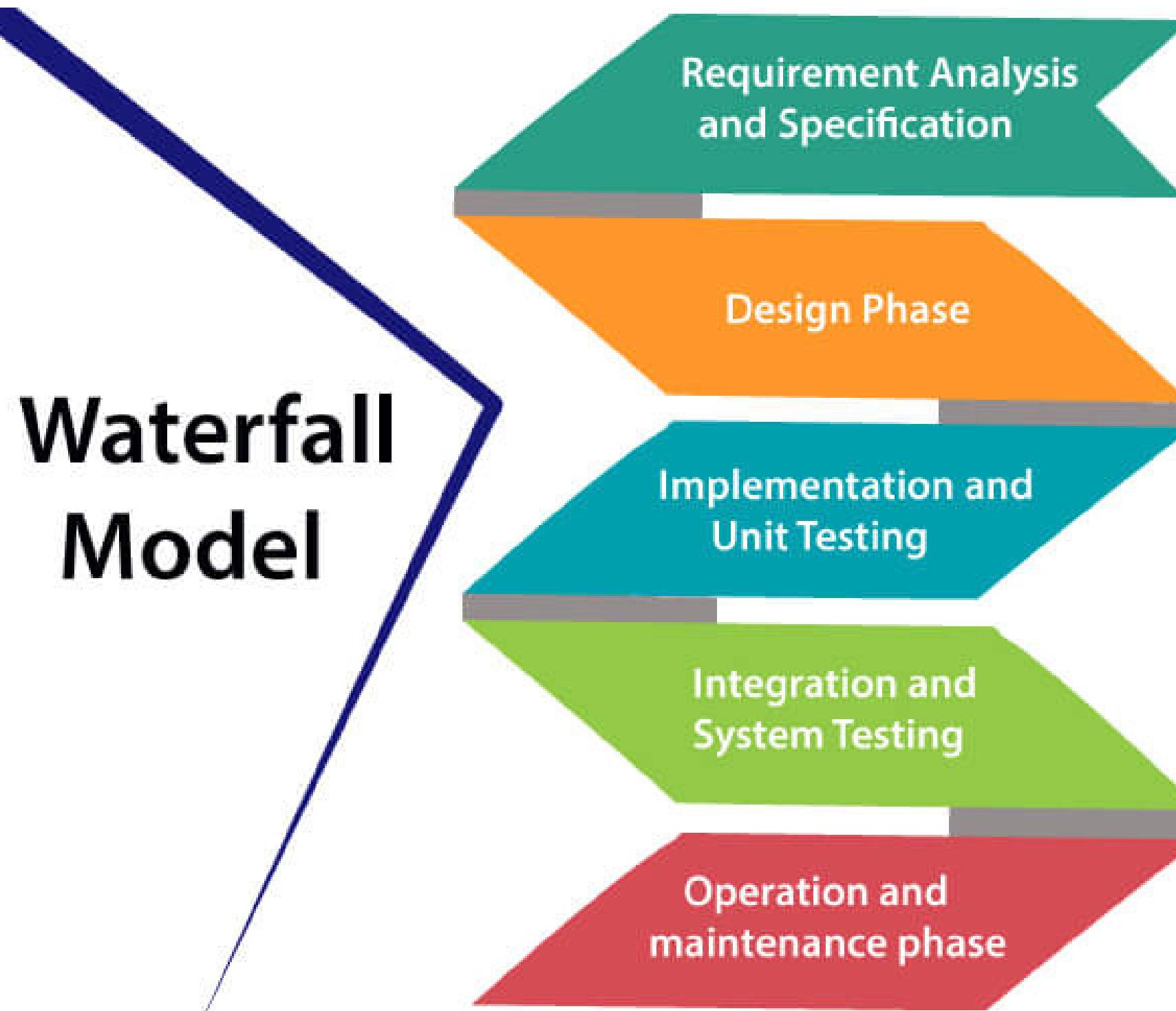


## SDLC (Models)



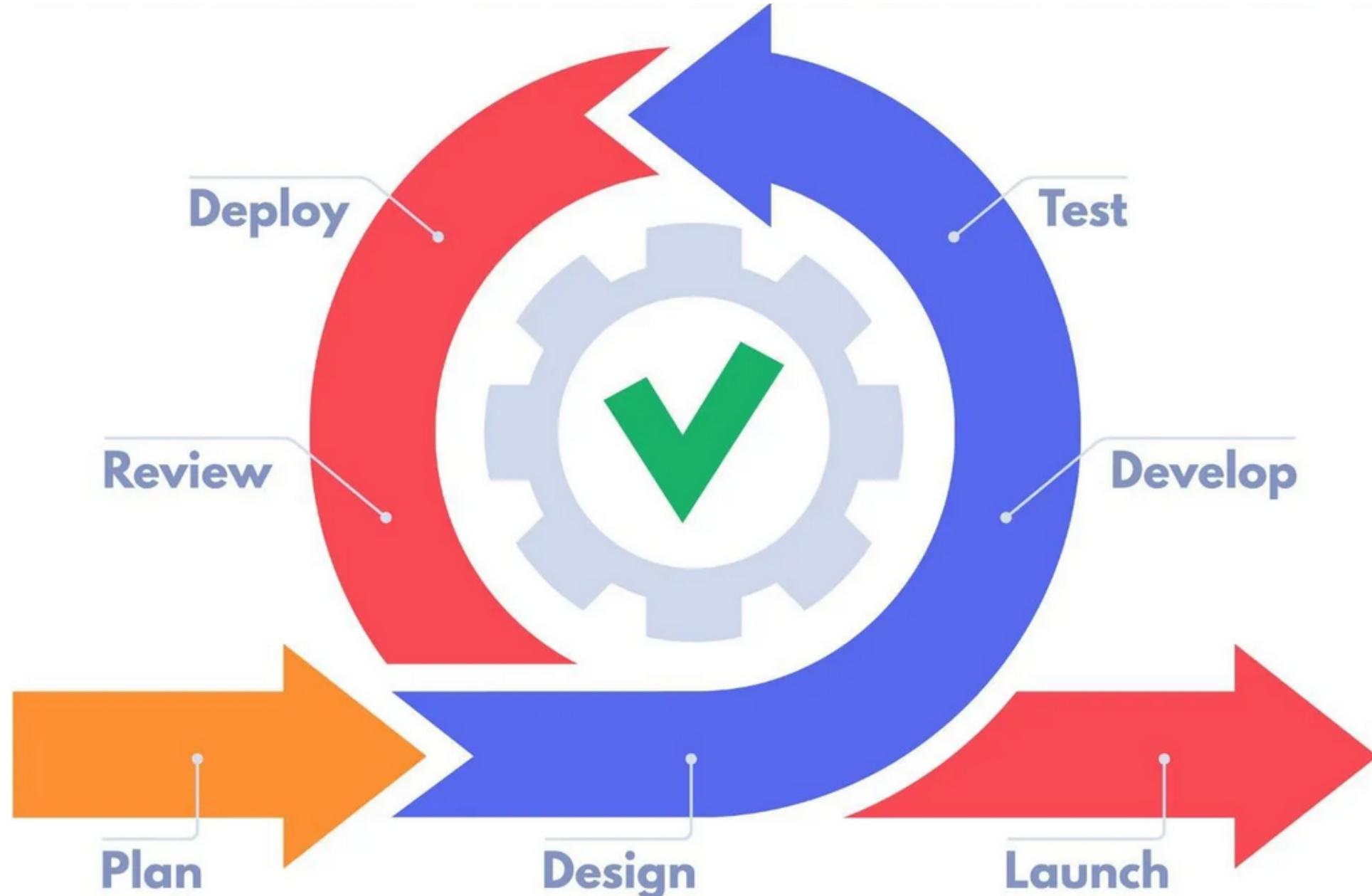
# SDLC Models

There are different software development life cycle models specify and design, which are followed during the software development phase.



# Waterfall model

The developer must complete every phase before the next phase begins. This model is named "Waterfall Model", because its diagrammatic representation resembles a cascade of waterfalls.



# Agile Model

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.

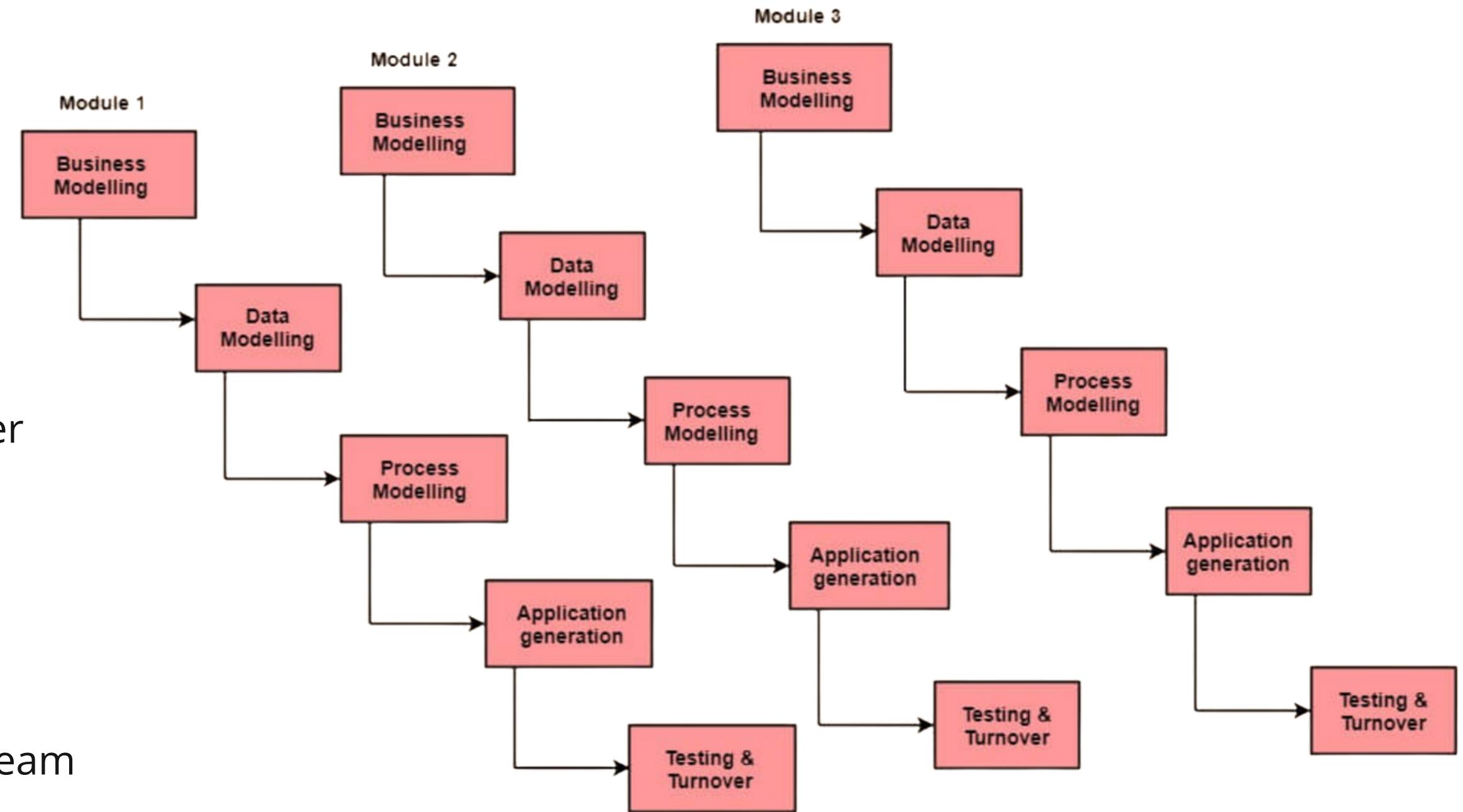
Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.



# RAD

RAD (Rapid Application Development) is a concept that products can be developed faster and of higher quality

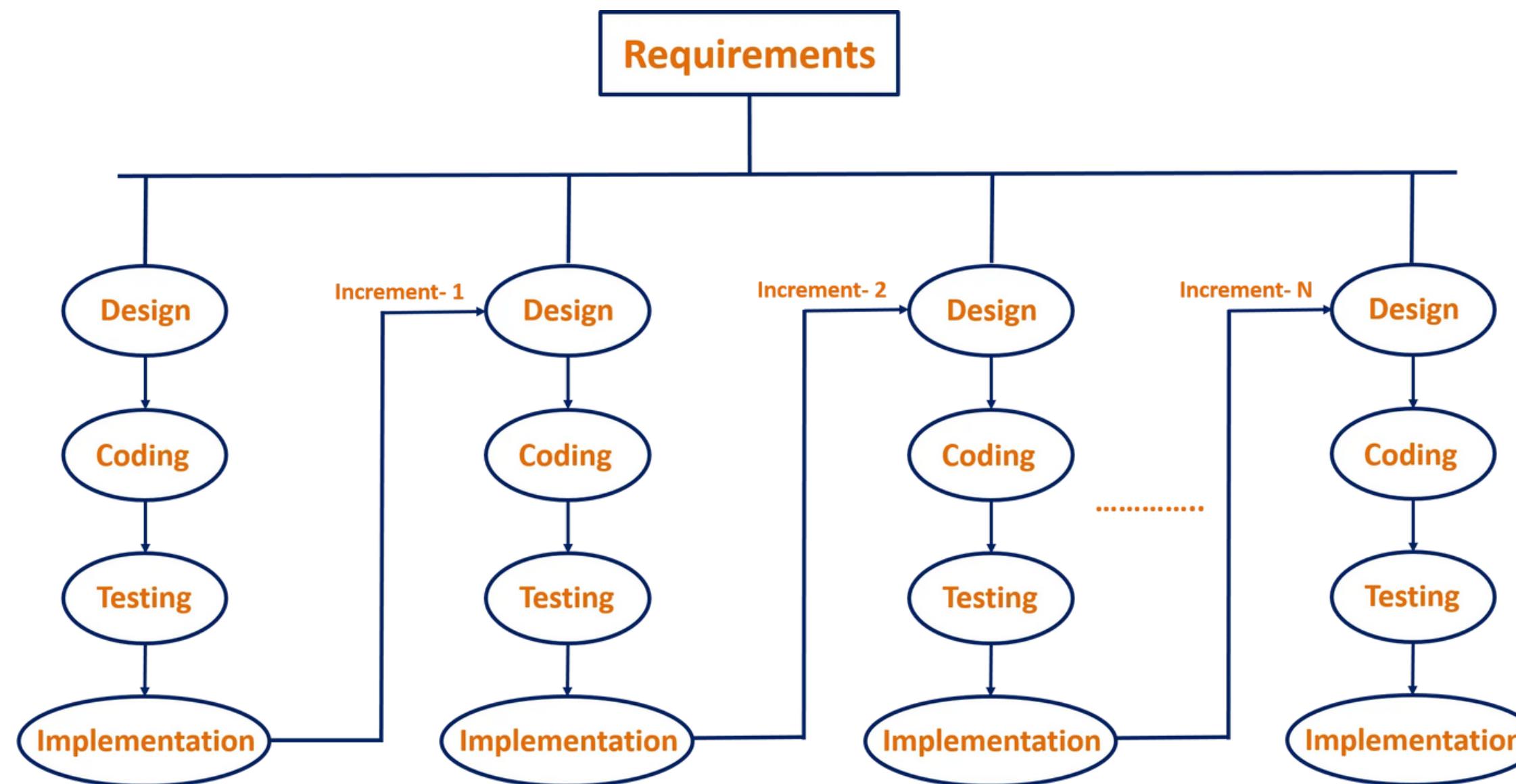
- Gathering requirements using workshops or focus groups
- Prototyping and early, reiterative user testing of designs
- The re-use of software components
- A rigidly paced schedule that refers design improvements to the next product version
- Less formality in reviews and other team communication





# Incremental Model

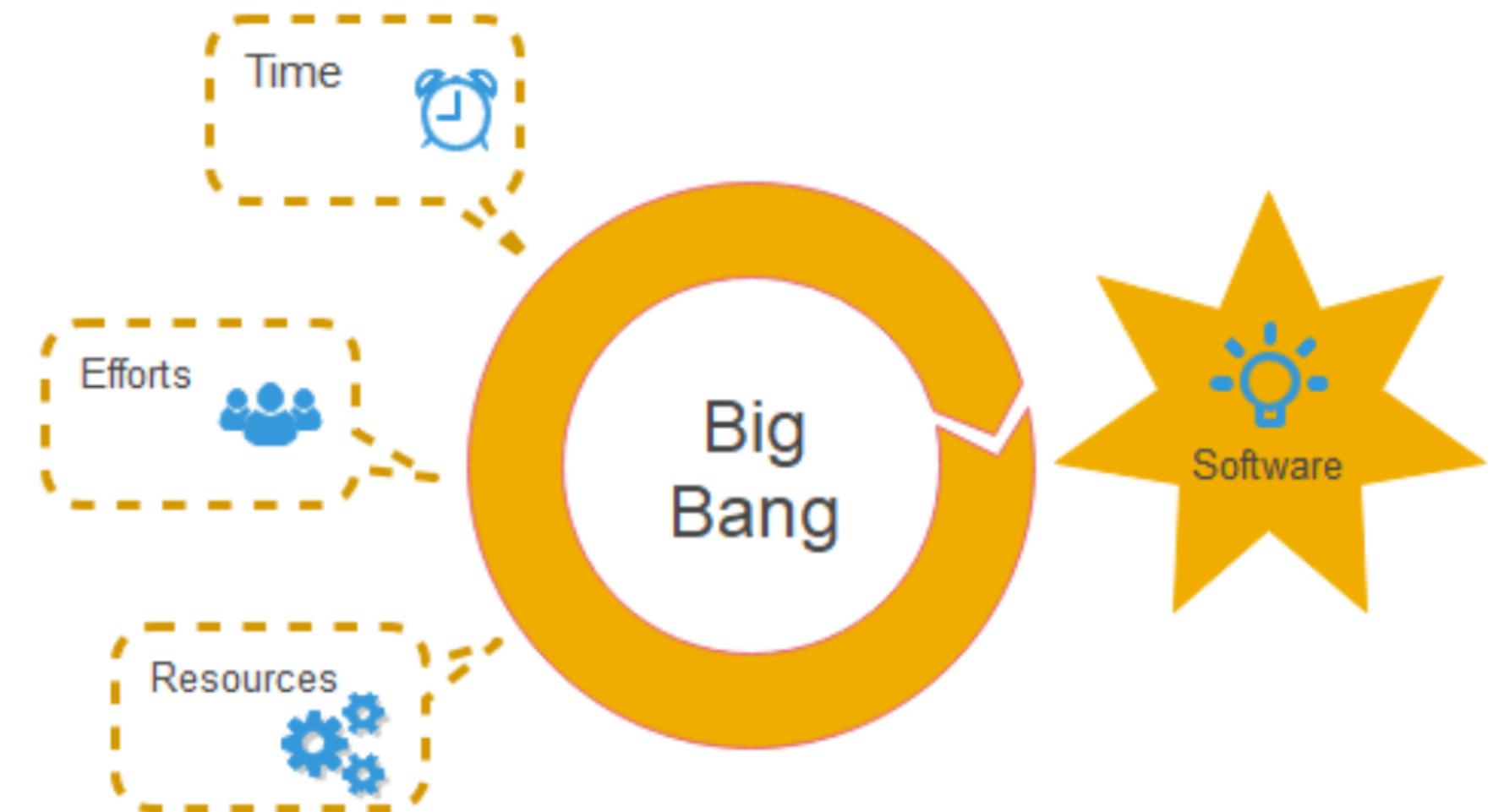
Incremental Model is a process of software development where requirements divided into multiple standalone modules of the software development cycle.





# Big Bang Model

- In this model, developers do not follow any specific process.
- Development begins with the necessary funds and efforts in the form of inputs.
- This model is ideal for small projects like academic projects or practical projects





LEARN WITH RABBIL

# Software project management

It is a procedure of managing, allocating and timing resources to develop computer software that fulfills requirements. There are three needs for software project management-

01 Time

02 Cost

03 Quality





# Project Manager

A project manager is a character who has the overall responsibility for the planning, design, execution, monitoring, controlling and closure of a project. A project manager represents an essential role in the achievement of the projects.

## Leader

A project manager must lead his team and should provide them direction to make them understand what is expected from all of them.

## Medium

The Project manager is a medium between his clients and his team. He must coordinate and transfer all the appropriate information from the clients to his team and report to the senior management.

## Mentor

He should be there to guide his team at each step and make sure that the team has an attachment. He provides a recommendation to his team and points them in the right direction.



# Responsibilities of a Project Manager

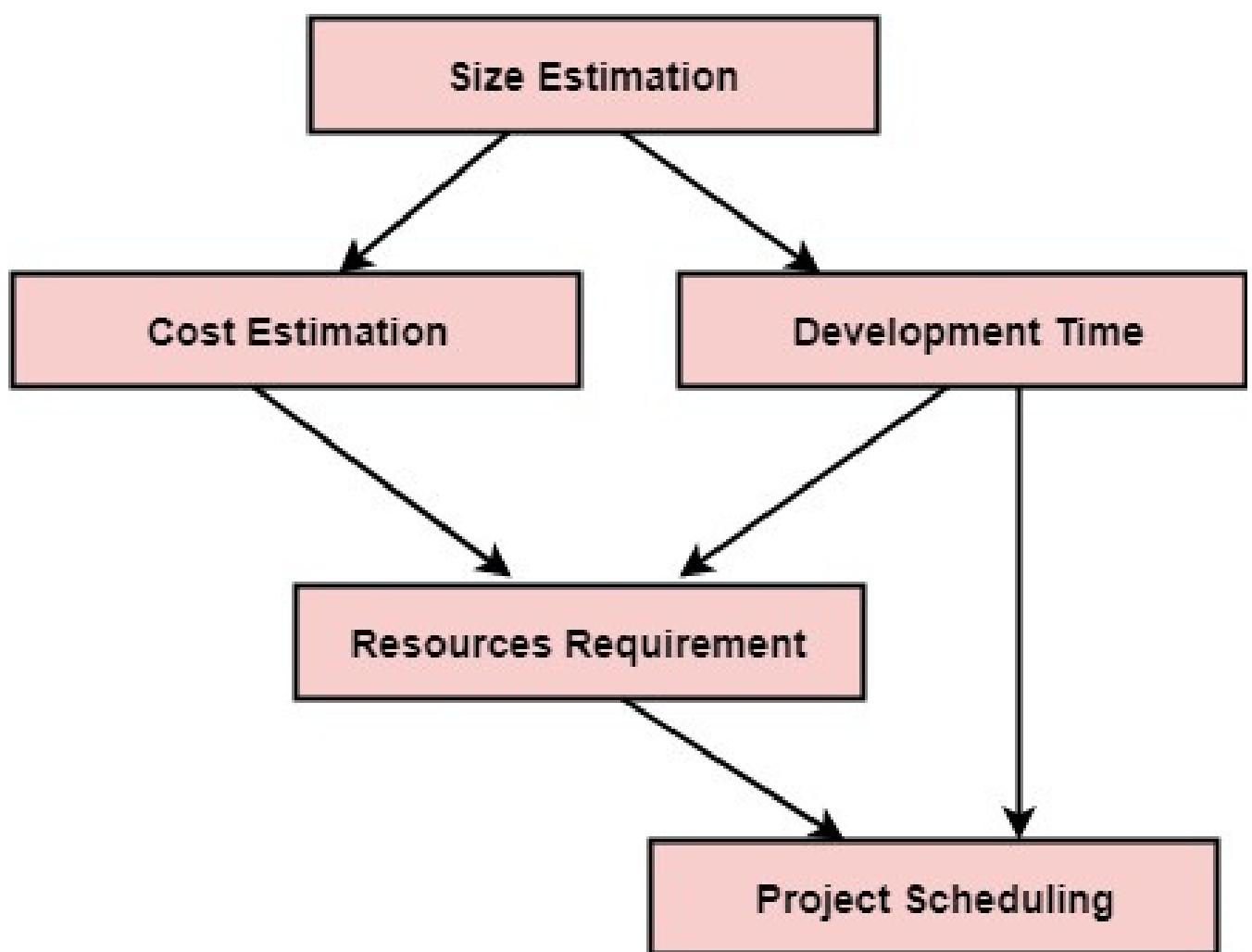
- Managing risks and issues
- Create the project team and assigns tasks
- Activity planning and sequencing
- Monitoring and reporting progress
- Modifies the project plan to deal with the situation





# Software Project Planning

Software manager is responsible for planning and scheduling project development. Software Project planning starts before technical work start. The various steps of planning activities are





LEARN WITH RABBIL

# Risk Management

A software project can be concerned with a large variety of risks. In order to be adept to systematically identify the significant risks which might affect a software project, it is essential to classify risks into different classes.



## Project risks

Project risks concern different forms of budgetary, schedule, personnel, resource, and customer-related problems.

## Technical risks

Technical risks concern potential method, implementation, interfacing, testing, and maintenance issues.

## Business risks

This type of risks contain risks of building an excellent product that no one needs, losing budgetary or personnel commitments, etc.

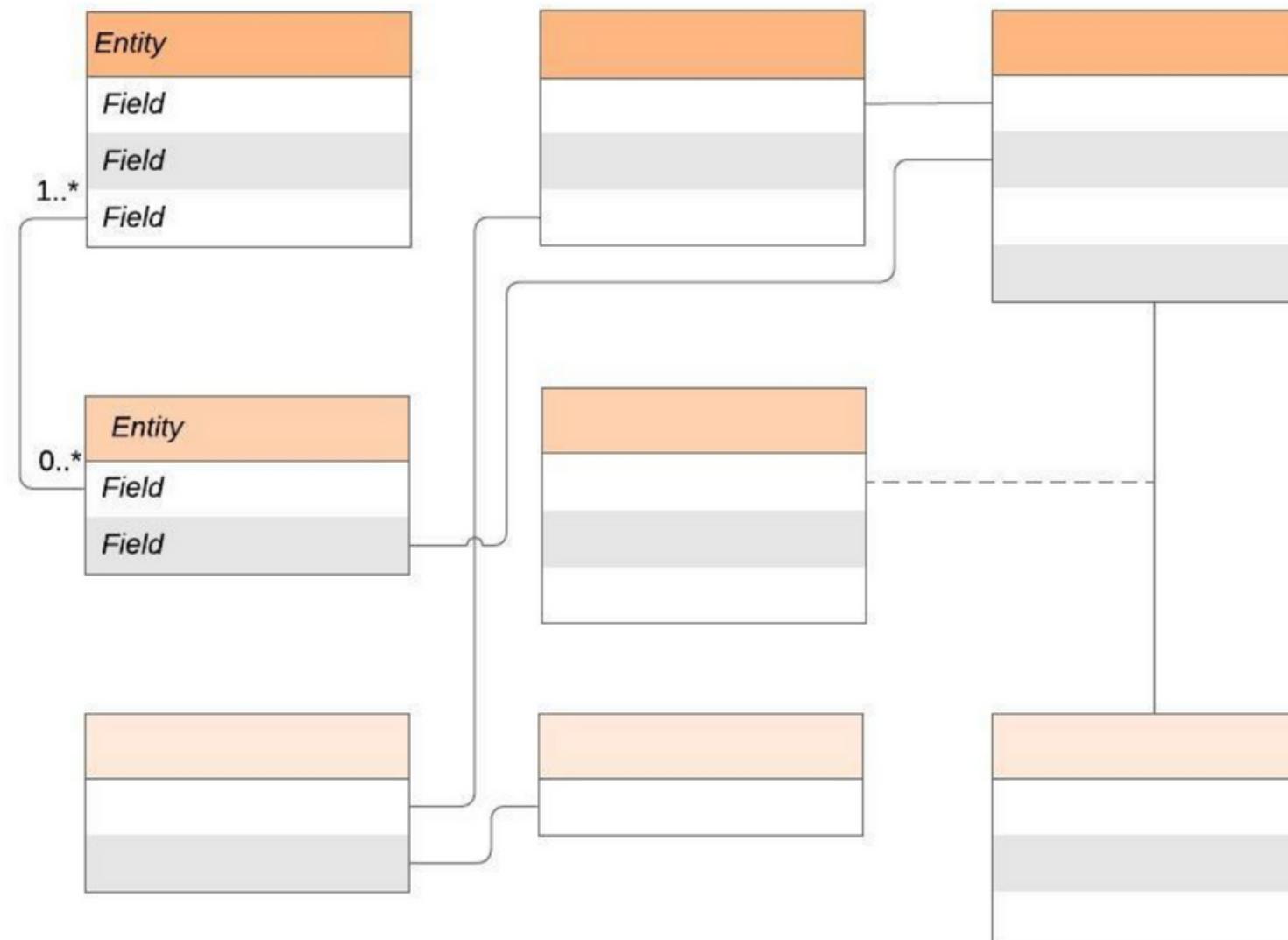


- Identify all the functions required to complete the project.
- Break down large functions into small activities.
- Determine the dependency among various activities.
- Establish the most likely size for the time duration required to complete the activities.
- Allocate resources to activities.
- Plan the beginning and ending dates for different activities.
- Determine the critical path.

# Project Scheduling

Project-task scheduling is a significant project planning activity. It comprises deciding which functions would be taken up when. To schedule the project plan, a software project manager wants to do the following:





# Entity-Relationship Diagrams

ER-modeling is a data modeling method used in software engineering to produce a conceptual data model of an information system.

- The database analyst gains a better understanding of the data to be contained in the database through the step of constructing the ERD.
- The ERD serves as a documentation tool
- Finally, the ERD is used to connect the logical structure of the database to users. In particular, the ERD effectively communicates the logic of the database to users.



# Software Design Stage

- Understanding project requirements
- Research and Analysis
- Design
- Prototyping
- Evaluation





# Software Design

## Stage 01

### Understanding project requirements

- Look up on projects features
- Project mission vision goals
- User Demands
- Ins & Outs of business document





LEARN WITH RABBIL

# Software Design

## Stage 02

### Research and Analysis

- Interviews
- Focus groups
- Survey





# Software Design

## Stage 03

### Design

- Wireframing
- Creating user stories
- Data flow diagrams
- Technical Design
- User Interface



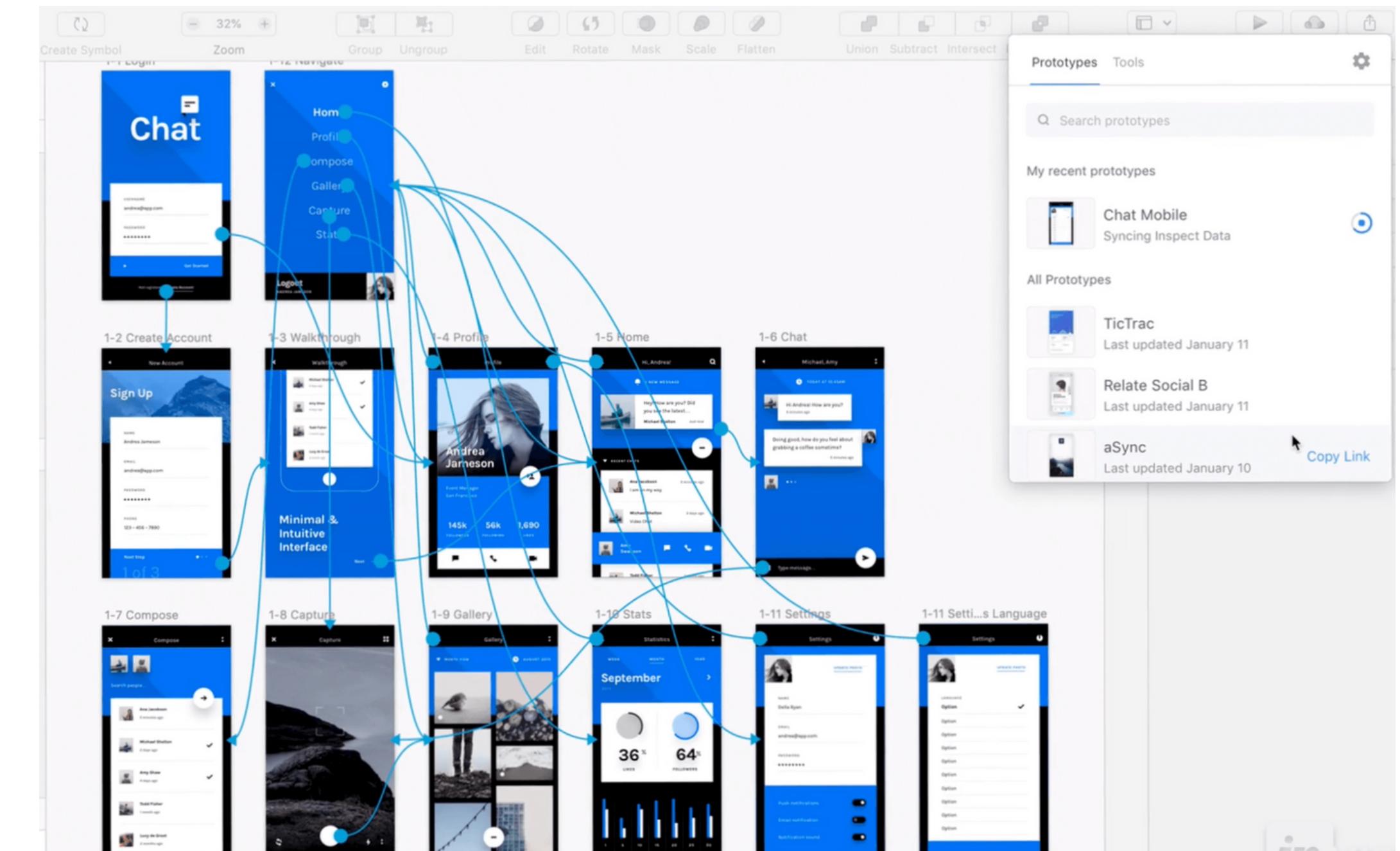


LEARN WITH RABBIL

# Software Design Stage 04

## Prototyping

- Low Fidelity Prototyping
- Medium Fidelity Prototyping
- High Fidelity Prototyping





# Software Design

## Stage 05

### Evaluation

#### Correctness

Software design should be correct as per requirement.

#### Completeness

The design should have all components like data structures, modules, and external interfaces, etc

#### Efficiency

Resources should be used efficiently by the program.

#### Flexibility

Able to modify on changing needs.

#### Consistency

There should not be any inconsistency in the design

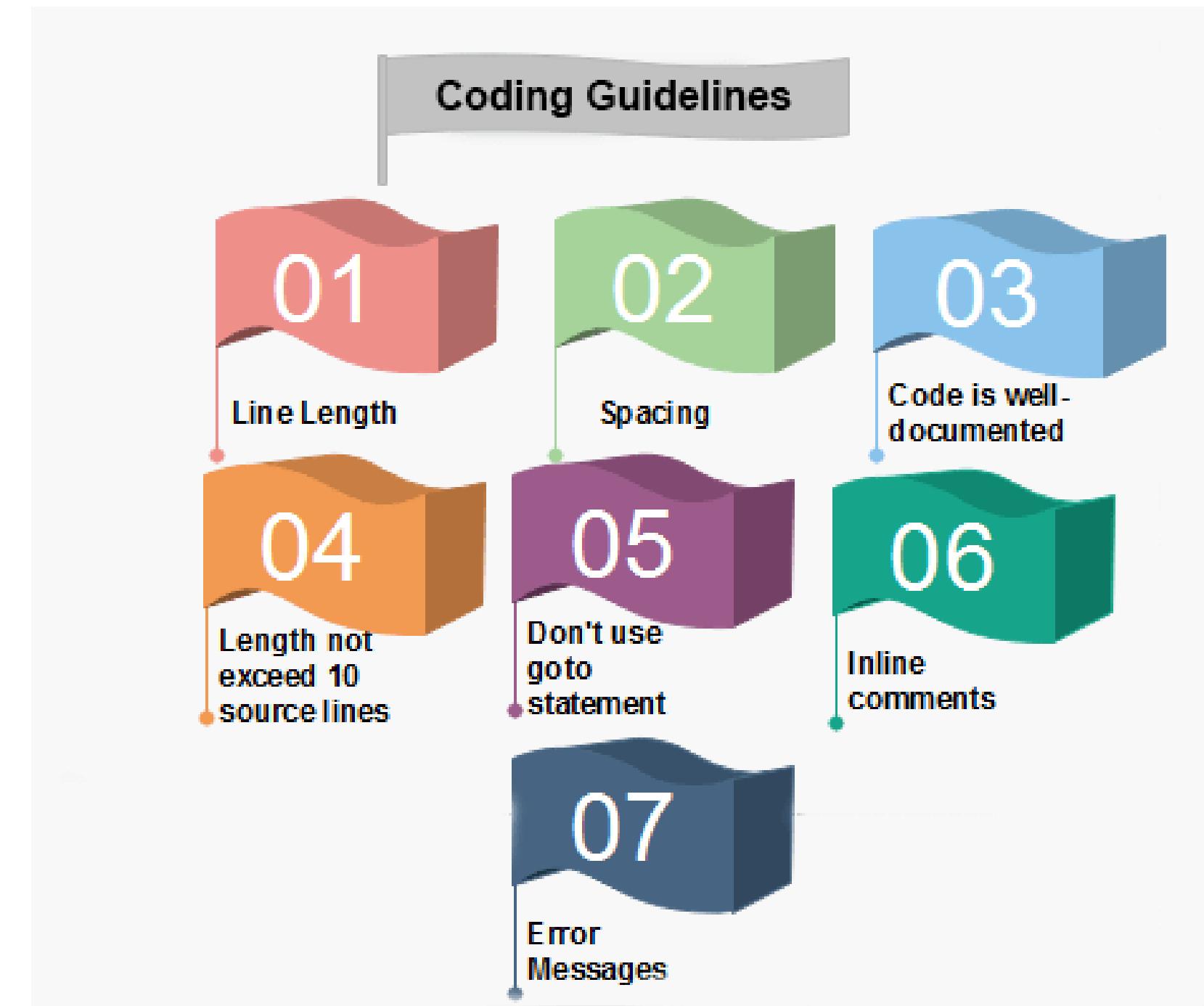
#### Maintainability

The design should be so simple so that it can be easily maintainable by other designers



# Coding

- The coding is the process of transforming the design of a system into a computer language format
- This coding phase of software development is concerned with software translating design specification into the source code.
- It is necessary to write source code & internal documentation so that conformance of the code to its specification can be easily verified.
- Coding is done by the coder or programmers who are independent people than the designer.
- The goal is not to reduce the effort and cost of the coding phase, but to cut to the cost of a later stage.
- The cost of testing and maintenance can be significantly reduced with efficient coding.





# Goals of Coding

01

## To translate the design of system into a computer language format

The coding is the process of transforming the design of a system into a computer language format, which can be executed by a computer and that perform tasks as specified by the design of operation during the design phase.

02

## To reduce the cost of later phases

The cost of testing and maintenance can be significantly reduced with efficient coding.

03

## Making the program more readable

Program should be easy to read and understand. It increases code understanding having readability and understandability as a clear objective of the coding activity can itself help in producing more maintainable software.



# Software Quality Assurance (SQA)

**Software quality assurance focuses on**

- software's portability
- software's usability
- software's reusability
- software's correctness
- software's maintainability
- software's error control





# Benefits of Software Quality Assurance (SQA)

- SQA produces high quality software.
- High quality application saves time and cost.
- SQA is beneficial for better reliability.
- SQA is beneficial in the condition of no maintenance for a long time.
- High quality commercial software increase market share of company.
- Improving the process of creating software.
- Improves the quality of the software.





# Software Maintenance

Software maintenance is a part of the Software Development Life Cycle. Its primary goal is to modify and update software application after delivery to correct errors and to improve performance. Software is a model of the real world. When the real world changes, the software require alteration wherever possible.

- Correct errors
- Change in user requirement with time
- Changing hardware/software requirements
- To improve system efficiency
- To optimize the code to run faster
- To modify the components
- To reduce any unwanted side effects.

