

# **Software Engineering Syllabus Overview**

# Syllabus Structure

**Savitribai Phule Pune University**  
**Second Year of Engineering (2019 Course)**  
**210253: Software Engineering**

Teaching Scheme	Credit Scheme	Examination Scheme and Marks
Lecture: <b>03 Hours/Week</b>	<b>03</b>	Mid_Semester(TH): <b>30 Marks</b> End_Semester(TH): <b>70 Marks</b>

**Prerequisite Courses :** 110005: Programming and Problem Solving

**Companion Course :** ---

## **Course Objectives:**

The main objective of this course is to introduce the students to software engineering- the fundamentals of software engineering principles and practices, including project management, configurations management, requirements definition, system analysis, design, testing, and deployment with hands-on experience in a group software development project.

- To learn and understand the principles of Software Engineering.
- To be acquainted with methods of capturing, specifying, visualizing and analyzing software requirements.
- To apply design and testing principles to software project development.
- To understand project management through life cycle of the project.

## **Course Outcomes:**

On completion of the course, learner will be able to-

**CO1: Analyze** software requirements and formulate design solution for a software.

**CO2: Design** applicable solutions in one or more application domains using software engineering approaches that integrate ethical, social, legal and economic concerns.

**CO3: Apply** new software models, techniques and technologies to bring out innovative and novelistic solutions for the growth of the society in all aspects and evolving into their continuous professional development.

# Introduction of Unit Contents

Course Contents		
Unit I	Introduction to Software Engineering and Software Process Models	(06Hours)
<b>Software Engineering Fundamentals:</b> Introduction to software engineering, The Nature of Software, Defining Software, Software Engineering Practice. <b>Software Process:</b> A Generic Process Model, defining a Framework Activity, Identifying a Task Set, Process Patterns, Process Assessment and Improvement, Prescriptive Process Models, The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Concurrent Models, A Final Word on Evolutionary Processes. Unified Process, Agile software development: Agile methods, plan driven and agile development.		
<u>#Exemplar/Case Studies</u>	Agile Tools- JIRA	
<u>*Mapping of Course Outcomes for Unit I</u>	CO1, CO3, CO7	
Unit II	Software Requirements Engineering and Analysis	(07 Hours)

**Modeling:** Requirements Engineering, Establishing the Groundwork, Identifying Stakeholders, Recognizing Multiple Viewpoints, working toward Collaboration, Asking the First Questions, Eliciting Requirements, Collaborative Requirements Gathering, Usage Scenarios, Elicitation Work Products, Developing Use Cases, Building the Requirements Model, Elements of the Requirements Model, Negotiating Requirements, Validating Requirements.



Unit III	Estimation and Scheduling	(07 Hours)
<b>Estimation for Software Projects:</b> The Project Planning Process, Defining Software Scope and Checking Feasibility, Resources management, Reusable Software Resources, Environmental Resources, Software Project Estimation, Decomposition Techniques, Software Sizing, Problem-Based Estimation, LOC-Based Estimation, FP-Based Estimation, Object Point (OP)-based estimation, Process-Based Estimation, Process-Based Estimation, Estimation with Use Cases, Use-Case–Based Estimation, Reconciling Estimates, Empirical Estimation Models, The Structure of Estimation Models, The COCOMO II Mode, Preparing Requirement Traceability Matrix <b>Project Scheduling:</b> Project Scheduling, Defining a Task for the Software Project, Scheduling. <b>Suggested Free Open Source Tools:</b> Gantt Project, Agantty, Project Libre.		
<a href="#">#Exemplar/Case Studies</a>	Write SRS in IEEE format for selected Project Statement/ case study, Study SRS of Online Voting system, Library management System ( <a href="http://dos.iitm.ac.in/OOSD_Material/CaseStudies/CaseStudy2/eVote-srs.pdf">http://dos.iitm.ac.in/OOSD_Material/CaseStudies/CaseStudy2/eVote-srs.pdf</a> ),	
<a href="#">*Mapping of Course Outcomes for Unit III</a>	CO1, CO3, CO7	
Unit IV	Design Engineering	(07 Hours)
<b>Design Concepts:</b> Design within the Context of Software Engineering, The Design Process, Software Quality Guidelines and Attributes, Design Concepts - Abstraction, Architecture, design Patterns, Separation of Concerns, Modularity, Information Hiding, Functional Independence, Refinement, Aspects, Refactoring, Object-Oriented Design Concept, Design Classes, The Design Model , Data Design Elements, Architectural Design Elements, Interface Design Elements, Component-Level Design Elements, Component Level Design for Web Apps, Content Design at the Component Level, Functional Design at the Component Level, Deployment-Level Design Elements. <b>Architectural Design:</b> Software Architecture, What is Architecture, Why is Architecture Important, Architectural Styles, A brief Taxonomy of Architectural Styles. <b>Suggested Free Open Source Tool:</b> Smart Draw		
<a href="#">#Exemplar/Case Studies</a>	Study design of Biometric Authentication software	
<a href="#">*Mapping of Course Outcomes for Unit IV</a>	CO1,CO2 CO3, CO7	



Unit V	Risks and Configuration Management	(07 Hours)
<b>Risk Management:</b> Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring, and Management, The RMMM Plan. <b>Software Configuration Management:</b> Software Configuration Management, The SCM Repository The SCM Process, Configuration Management for any suitable software system. <b>Suggested Free Open Source Tools:</b> CF Engine Configuration Tool, Puppet Configuration Tool.		
<a href="#">#Exemplar/Case Studies</a>	Risk management in Food delivery software	

<http://collegecirculars.unipune.ac.in/sites/documents/Syllabus2020/Forms/AllItems.aspx>

#56/87

Curriculum for Second Year of Computer Engineering (2019 Course), Savitribai Phule Pune University

<a href="#">*Mapping of Course Outcomes for Unit V</a>	CO1,CO2 CO3, CO7	
Unit VI	Software Testing	(07 Hours)
A Strategic Approach to Software Testing, Verification and Validation, Organizing for Software Testing, Software Testing Strategy—The Big Picture, Criteria for Completion of Testing, Strategic Issues, Test Strategies for Conventional Software, Unit Testing, Integration Testing, Test Strategies for Object-Oriented Software, Unit Testing in the OO Context, Integration Testing in the OO Context, Test Strategies for WebApps, Validation Testing, Validation-Test Criteria, Configuration Review. <b>Suggested Free Open Source Tools:</b> Selenium, JUnit.		
<a href="#">#Exemplar/Case Studies</a>	Selemium Testing with any online application	

# Study References

## Text Books:

1. Roger Pressman, "Software Engineering: A Practitioner's Approach"||, McGraw Hill, ISBN 0-07-337597-7
2. Ian Sommerville, "Software Engineering"||, Addison and Wesley, ISBN 0-13-703515-2

## Reference Books:

1. Carlo Ghezzi, "Fundamentals of Software Engineering", PHI, ISBN-10: 0133056996
2. Rajib Mall, "Fundamentals of Software Engineering"||, PHI, ISBN-13: 978-8120348981
3. Pankaj Jalote, "An Integrated Approach to Software Engineering"||, Springer, ISBN 13: 9788173192715.
4. S K Chang, "Handbook of Software Engineering and Knowledge Engineering"||, World Scientific, Vol I, II, ISBN: 978-981-02-4973-1
5. Tom Halt, "Handbook of Software Engineering", Clanye International ISBN-10: 1632402939

# What is Software Engineering ?

- **Software:-** Set of instructions or set of programs are known as software.
  
- **Types of Software:-**
  1. System Software
  2. Application Software
  3. Utility Software

## Cont'd....

- **System Software:** It is a software designed to provide a platform to other software.

Ex. Operating System (Windows, Android, Linux)

- **Application Software:** The software that helps you to do a specific type of works is called as application software.

Ex. Ms word, Excel etc.....



## Cont'd....

- **Utility Software:** To manage, maintain and control computer resources.

Ex. Antivirus Software

# Characteristics of Software

- Software is developed or engineered and not manufactured.
- Software is custom built
- Software doesn't wear out

## Role of software

- i. Product
- ii. Process

# Cont'd....

- **Engineering:** Application of science, tools and methods to find out cost effective solution to problems.
- **Software Engineering:-** It is the method of applying scientific and technological knowledge, procedures and rules to design, develop, test and maintain the software the product.
- **OR** Systematic, disciplined, quantifiable approach for the development, operation, maintenance of s/w

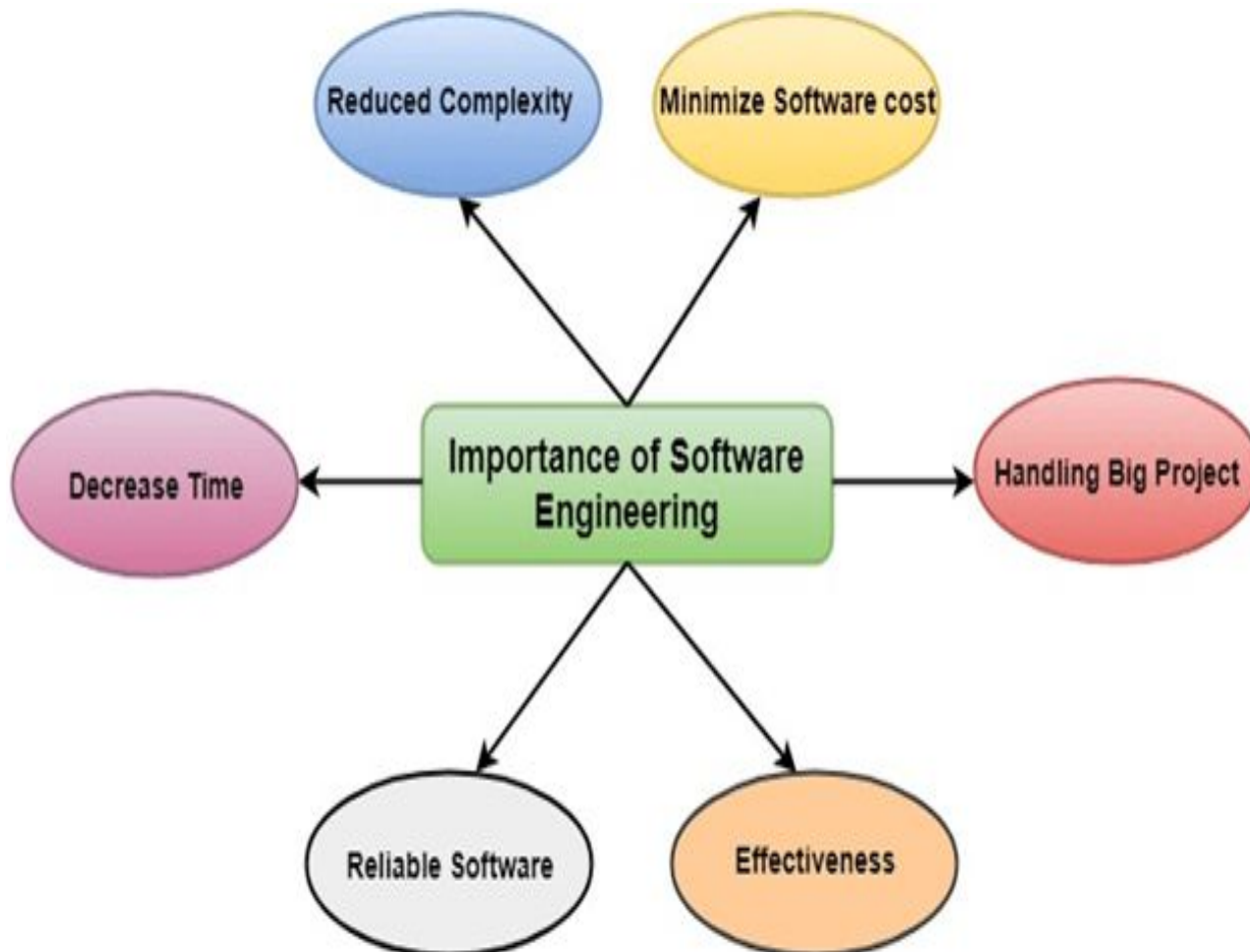


# Cont'd....

## Purpose of Software Engineering

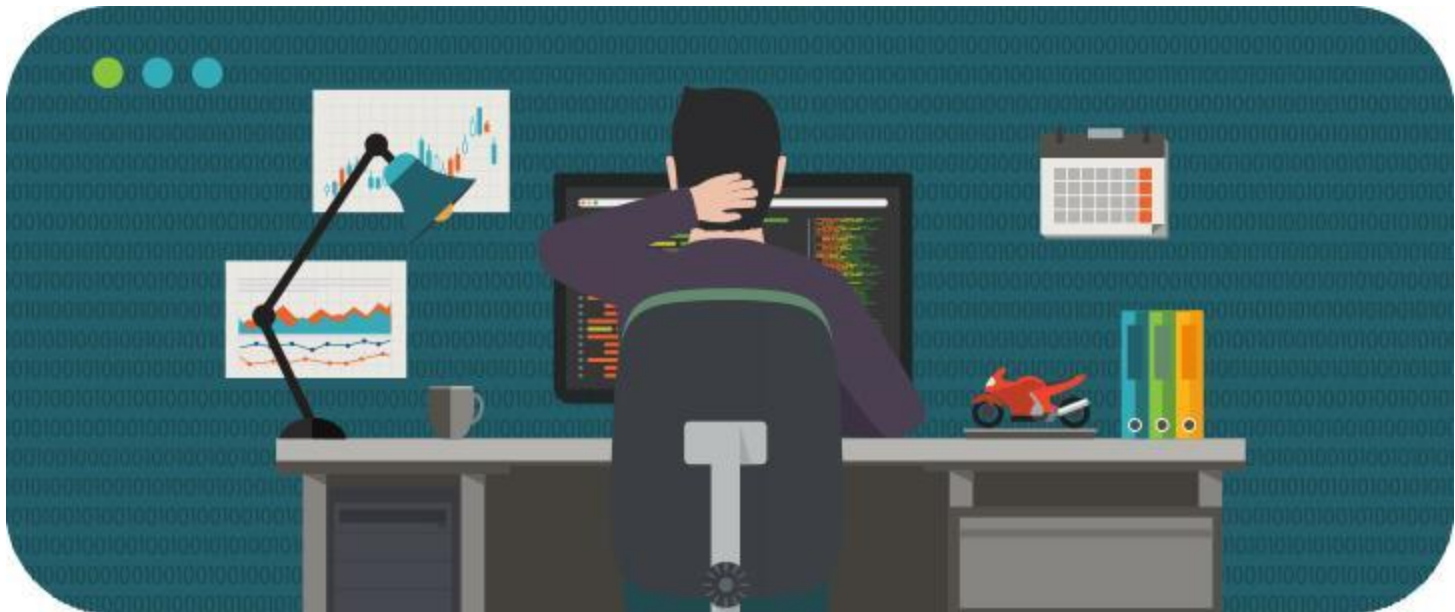
- To manage large software.
- To manage the cost.
- To manage the dynamic nature of the software.

# Importance of Software Engineering



# Why Study Software Engineering ?

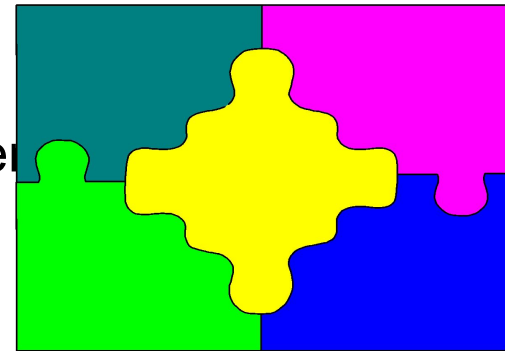
- To acquire skills to develop large programs.
  - Exponential growth in complexity and difficulty level with size.
  - The adhoc approach breaks down when size of software increases





# Cont'd....

- Ability to solve complex programming problems:
  - How to break large projects into smaller and manageable parts?
- To acquire skills to be a better programmer
  - Higher Productivity
  - Better Quality Programs
- Learn techniques of:
  - specification, design, interface development, testing, project management, etc.



# Nature of Software

- 7 broad categories of software:-
  1. System Software
  2. Application Software
  3. Engineering & Scientific Software
  4. Embedded Software
  5. Product-line software
  6. Web Applications
  7. Artificial Intelligence software

# Nature of Software

- 7 broad categories of software:-
  1. System Software
  2. Application Software
  3. Engineering & Scientific Software
  4. Embedded Software
  5. Product-line software
  6. Web Applications
  7. Artificial Intelligence software



## Cont'd....

- **System Software:-** It is a collection of program written to provide service for other program.

Example:- OS & compiler

- **Application Software:-** It is a program or group of programs designed for end users.

Example:- Email, spread sheets

## Cont'd....

- **Engineering & Scientific Software:-** The software is used to facilitate the engineering functions & tasks.

Example:- CAD

- **Embedded Software:-** It is used to control products & systems for consumer & industrial markets. It resides in ROM.
- **Example:-** GPS devices, calculators.

## Cont'd

- **Web applications:-** It is a client-server computer program which the client runs in a web browser.

Examples:- online auctions, web mails, websites

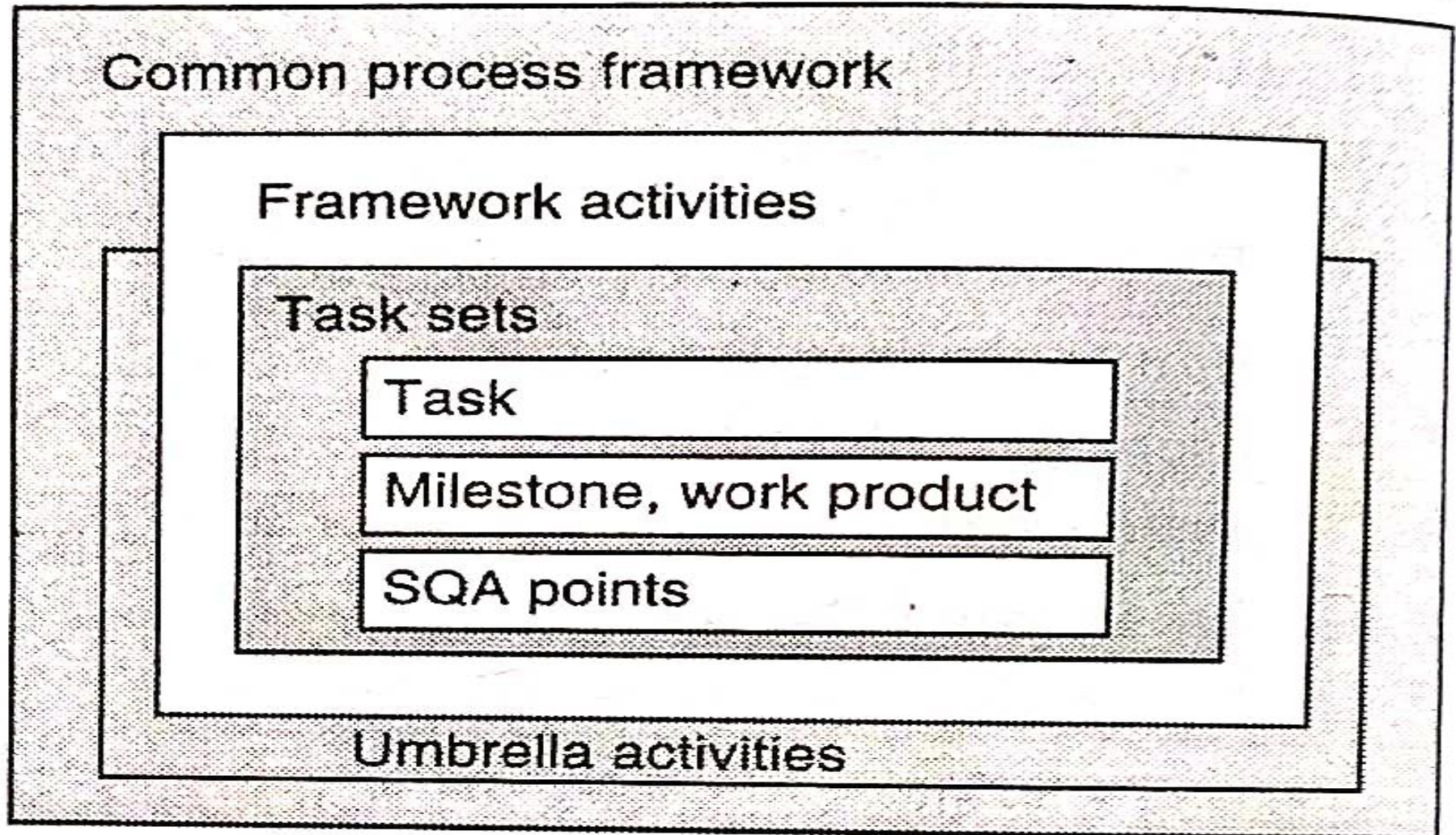
- **Artificial Intelligence software:-** ML learning includes algorithms that are developed to tell a computer how to respond to something by example.

Examples:- Speech recognition, Problem Solving  
Learning & planning

# Software Process

- **Process:-** A sequence of steps/actions taken to develop successful software system.
- **Process Framework:-** Establishes the for SE by identifying small no. of framework activities and umbrella activities that are applicable for entire software process.

# Software Process Framework





# Framework Activities

1. **Communication:-** involves communication with customer and gathering requirements.
2. **Planning:-** establishes plan for project
3. **Designing/ Modelling:-** constructing models so that customer can understand the requirements and design in a better way.

## Cont'd

4. **Construction/coding/implementation:-** combination of code generation and together.
5. **Deployment:-** Software is deliver to customer, Evaluate software and send feedback.

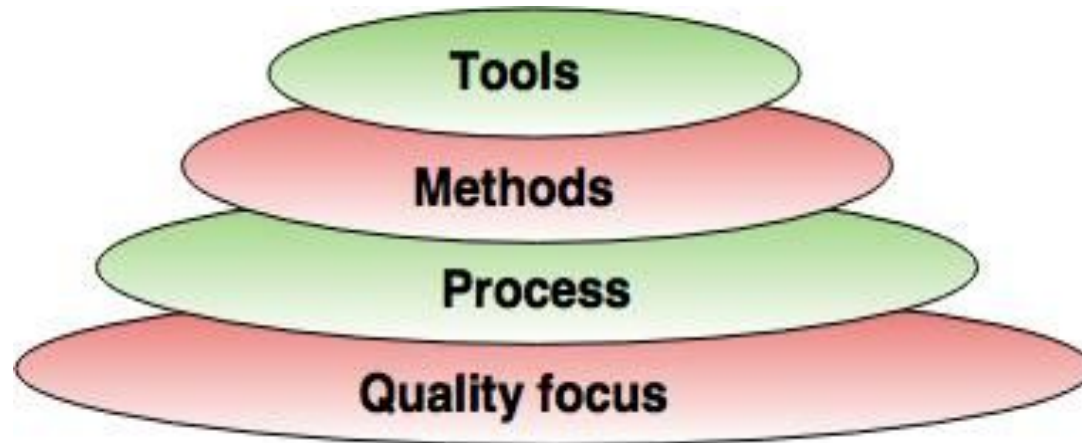
# Umbrella Activities

- **Framework activities:** Small projects
  - **Umbrella activities:** Large projects to track, control and manage the project
- 
1. **Software project tracking and control:-** Team
  2. **Risk management:-** Identify the risk, analyse, solve
  3. **Software quality Assurance:-** Testing the quality of the software.
  4. **Formal technical reviews:-** Modification done

## Cont'd....

5. Measurement:- Cost, time, manpower
6. Software configuration management:-  
Changes-effects
7. Reusability Management:- Criteria for reuse
8. Work product preparation and production:-  
Requirements to create work product

# SE Layered Architecture



**Fig. - Software Engineering Layers**



## Cont'd....

- Software engineering is a fully layered technology.
- To develop a software, we need to go from one layer to another.
- All these layers are related to each other and each layer demands the fulfillment of the previous layer.

# Cont'd....

## 1. Quality focus

**The characteristics of good quality software are:**

Correctness of the functions required to be performed by the software.

Maintainability of the software

Integrity i.e. providing security so that the unauthorized user cannot access information or data.

Usability i.e. the efforts required to use or operate the software.

## Cont'd....

**2. Process** : It is the base layer or foundation layer for the software engineering.

The software process is the key to keep all levels together.

It defines a framework that includes different activities and tasks.

In short, it covers all activities, actions and tasks required to be carried out for software development.

## Cont'd....

**3. Methods** :The method provides the answers of all 'how-to' that are asked during the process.

It provides the technical way to implement the software.

It includes collection of tasks starting from communication, requirement analysis, analysis and design modelling, program construction, testing and support.

## Cont'd....

4. **Tools** :The software engineering tool is an automated support for the software development. The tools are integrated i.e the information created by one tool can be used by the other tool.
- For example:** The Microsoft publisher can be used as a web designing tool.

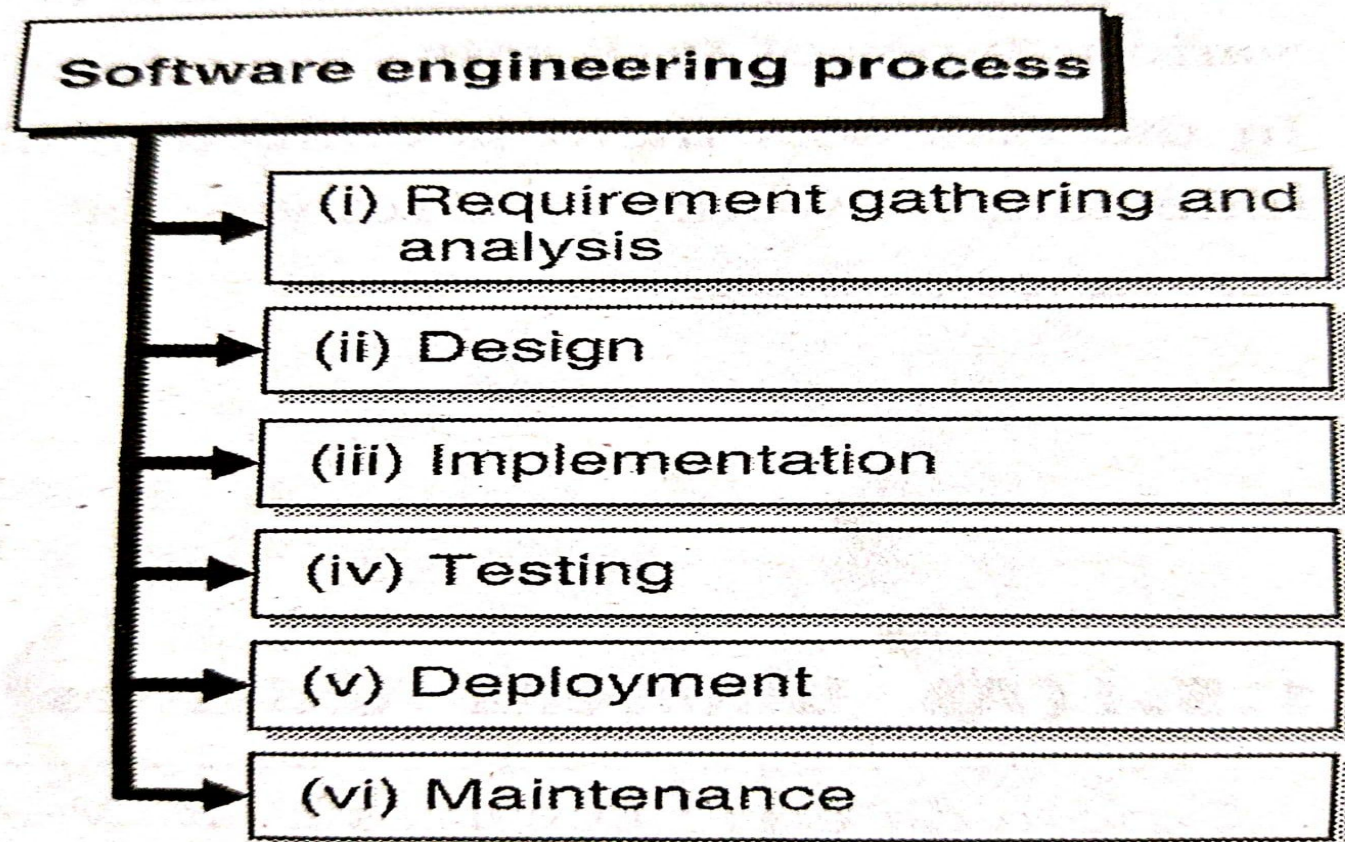
# Prescriptive Process Model

- A process model describes how each step in SDLC works & what are the process required for each step in SDLC.
  
- 5 Process Model
  1. Waterfall Model
  2. Incremental Model- a) Incremental b) RAD
  3. Iterative Model- a) Prototype b) Spiral
  4. Evolutionary Model
  5. Unified Model



# Software Development Life Cycle (SDLC)

- Refers to overall process involved in the s/w development.



# Cont'd....

## □ 6 Phases in SDLC :

1. Requirement gathering & analysis
2. Design
3. Implementation / development
4. Testing
5. Deployment
6. Maintenance

## Cont'd....

- **Requirement gathering & analysis:-** User requirements are collected from the user and are provided to development team & documents the requirements in the SRS document.
- **Design:-** Based on the requirement we make a design showing how the flow of process should happen.

Types - High level design e.g. Complete house  
- Low level design e.g. Features of every room

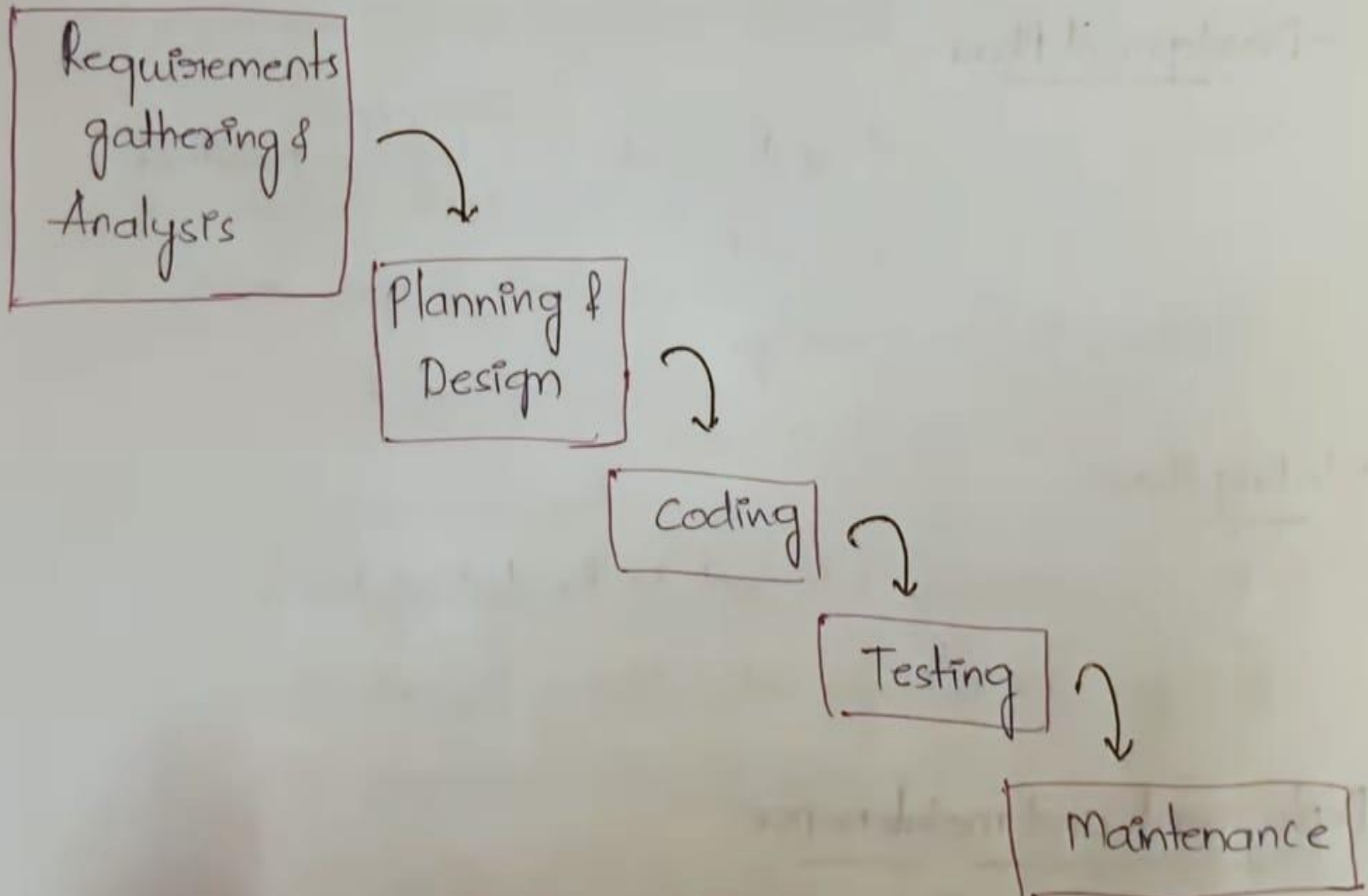
## Cont'd....

- **Implementation/ Development:-** We start writing the code for the s/w product. Output is code document.
- **Testing:-** Once the s/w is ready, it is sent to the testing team. Check errors, validation of Req. etc.
- **Deployment:-** Delivering the s/w to customer
- **Maintenance:-** Solving the problems of the customers.

# Waterfall Model

- ❑ Oldest Model
- ❑ Sequential Development
- ❑ Requirements
- ❑ No reverse direction
- ❑ Only when we are clear about requirements
- ❑ Low budget Projects
- ❑ No customer involvement
- ❑ Will not allow any changes

## Cont'd....





## Cont'd....

- The waterfall model is also called as 'Linear sequential model' or 'Classic life cycle model'.
- In this model, each phase is fully completed before the beginning of the next phase.
- This model is used for the small projects.
- In this model, feedback is taken after each phase to ensure that the project is on the right path.

# Cont'd....

## Advantages of Waterfall Model:

- The waterfall model is simple and easy to understand, implement, and use.
- All the requirements are known at the beginning of the project, hence it is easy to manage.

## Disadvantages of the Waterfall Model:

- This model is not good for complex and object oriented projects.
- It is a poor model for long projects.

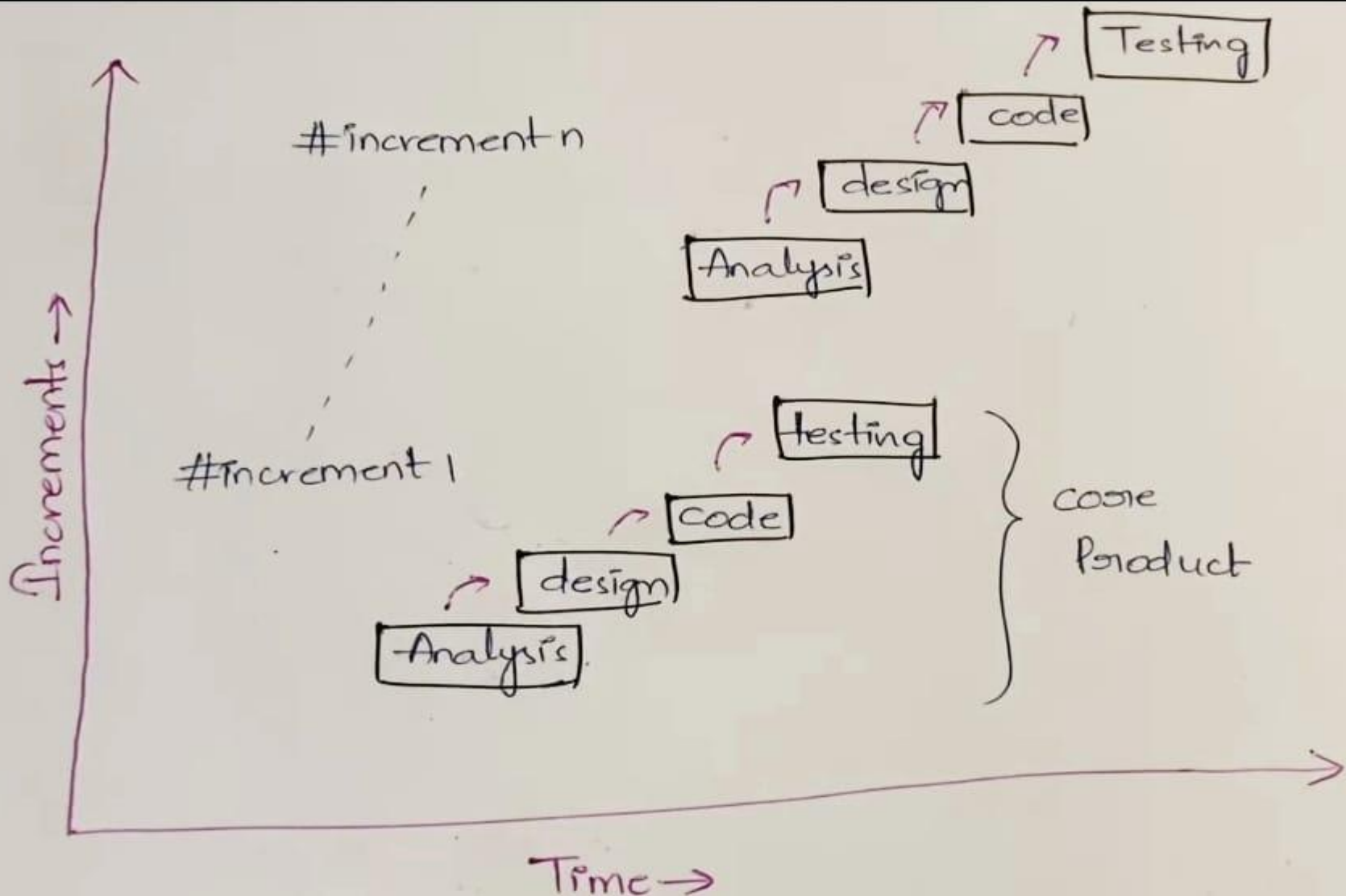
# Incremental Model

- Requirements are divided into multiple modules
- Each module – Analysis, design, code & Testing
- Used in day to day applications  
e.g. Banking
- Less manpower is required
- Two Models :-
  1. Incremental Model and RAD Model

# Incremental Model

- The incremental model combines the elements of waterfall model and they are applied in an iterative fashion.
- The first increment in this model is generally a core product.
- Each increment builds the product and submits it to the customer for any suggested modifications.
- The next increment implements on the customer's suggestions and add additional requirements in the previous increment.
- This process is repeated until the product is finished.

# Cont'd....



# Cont'd....

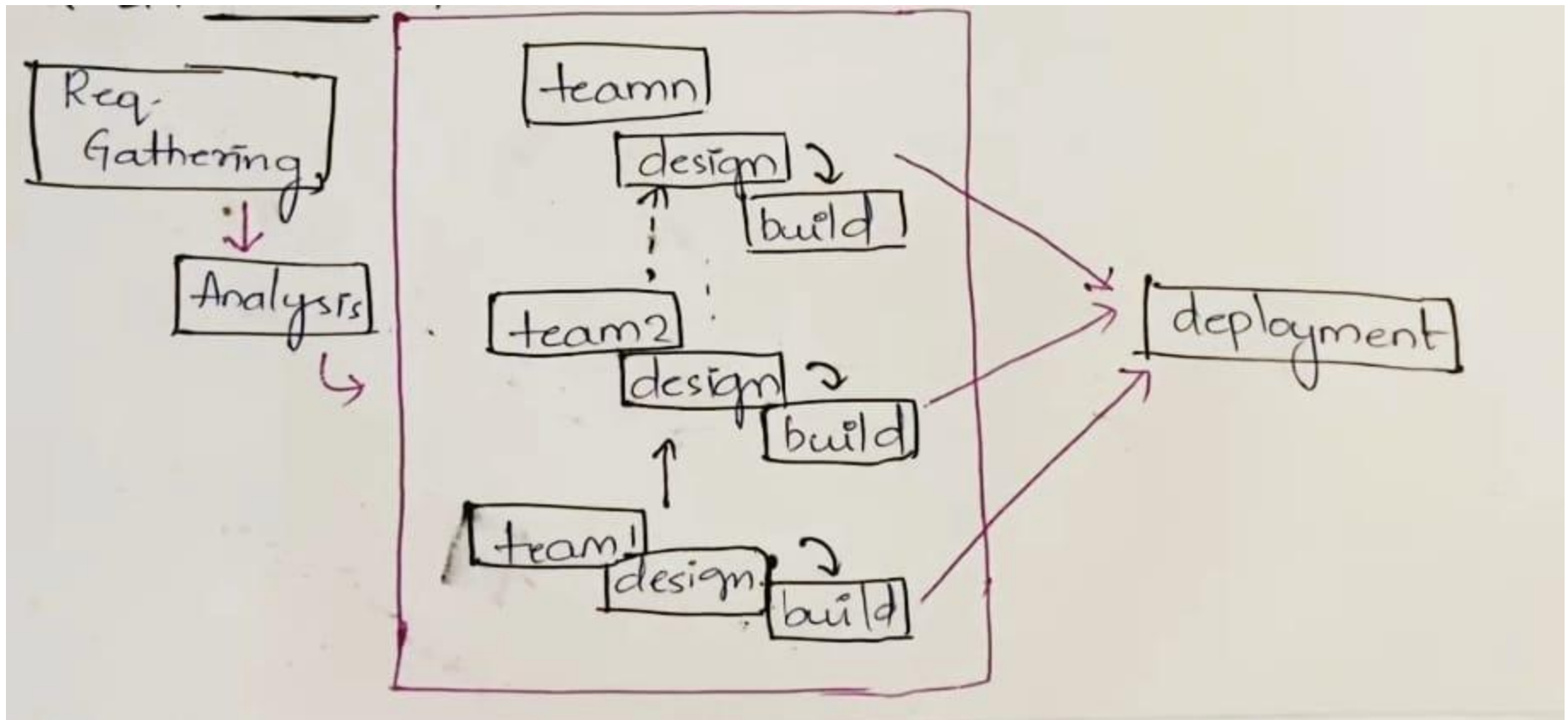
## Advantages of Incremental Model:

- This model is flexible because the cost of development is low and initial product delivery is faster.
- It is easier to test and debug during the smaller iteration.

## Disadvantages of the Incremental Model:

- The cost of the final product may cross the cost estimate initially.
- This model requires a very clear and complete planning.

# RAD Model



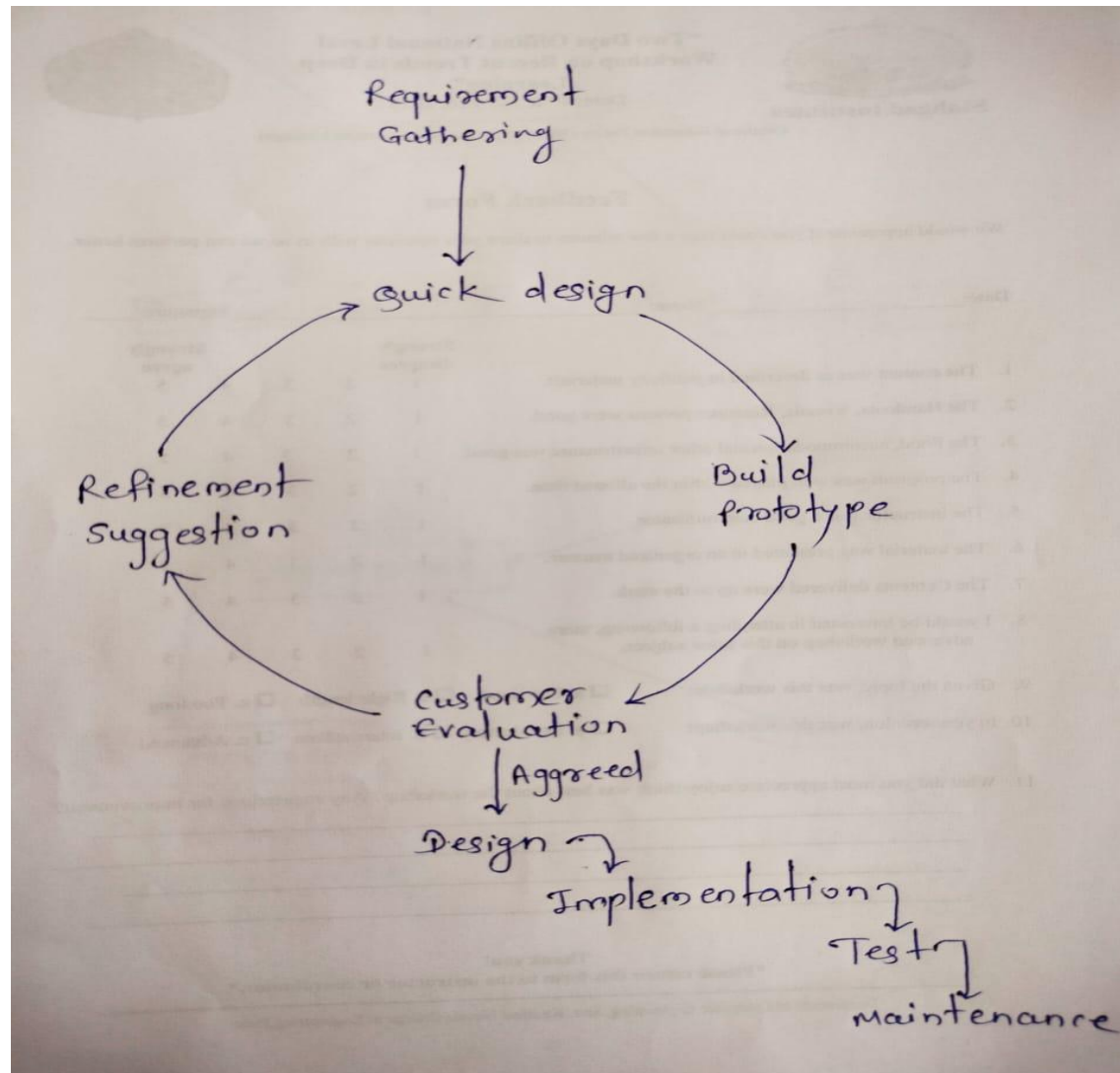


# Iterative Process Model

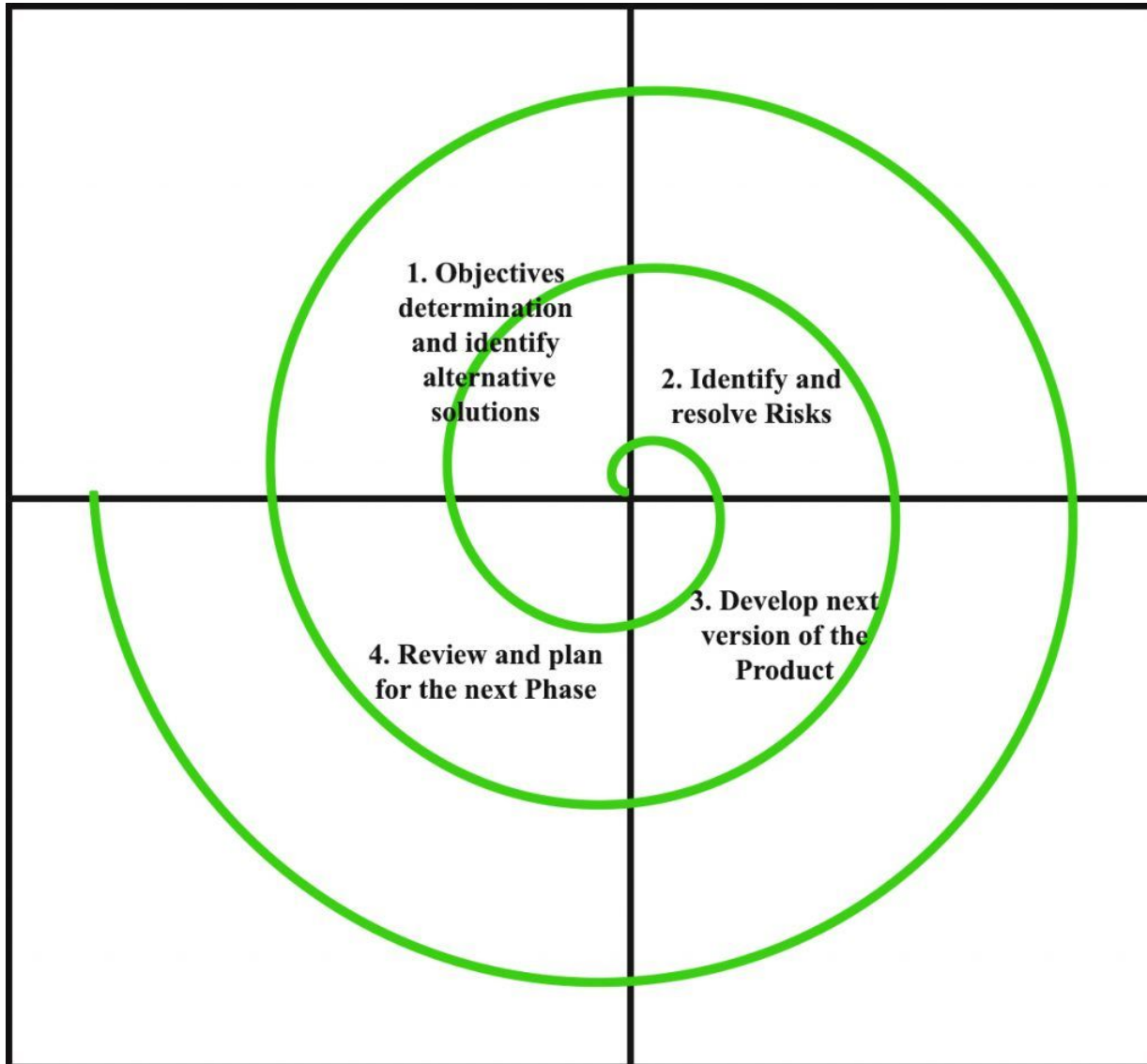
□ Two Types :-

1. Prototype Model:- Sample- dummy model
  - used when customer is not clear about req.
  - also called throwaway model
  
2. Spiral Model:- 4 stages
  - Repetitive activities
  - mainly used for risk handling

# Prototype Model



# Spiral Model



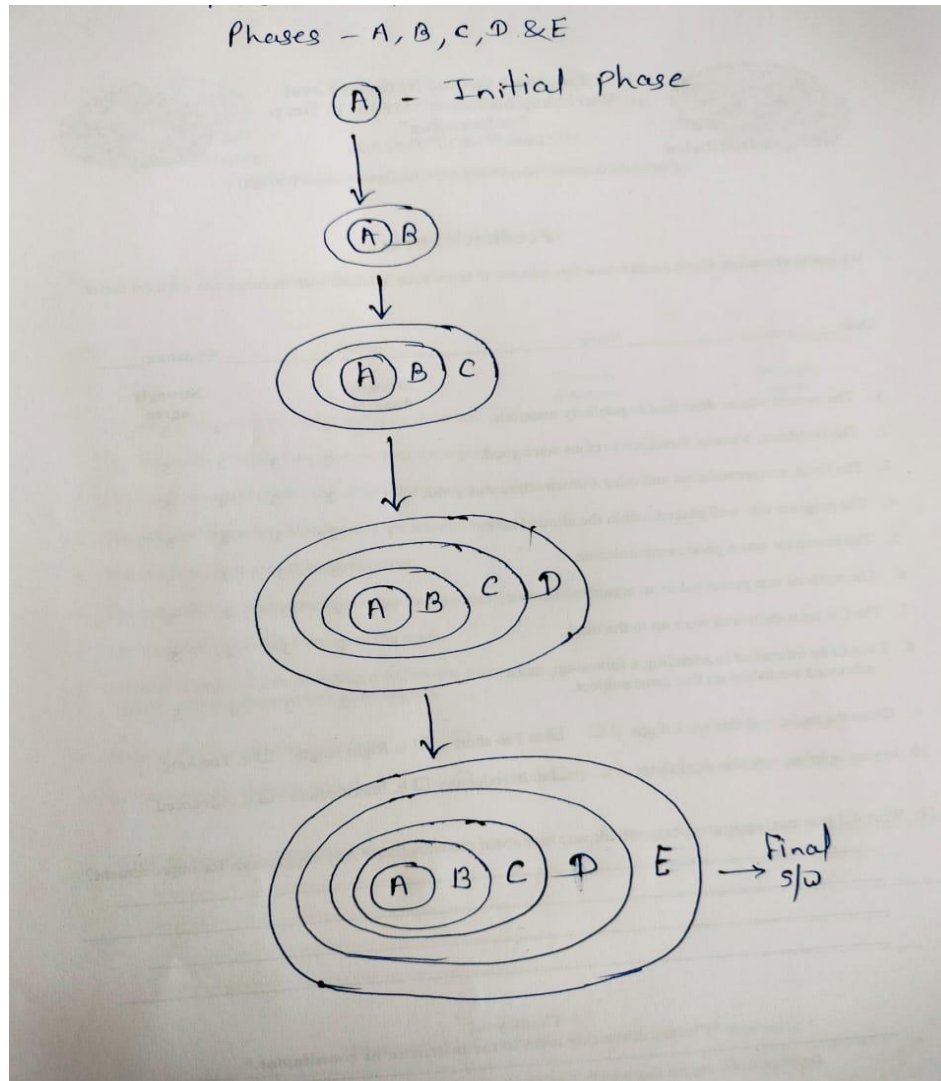
# Evolutionary Model

- Also called Successive Version Model
- Combination of iterative and incremental model
- Incremental model first implement a few basic features(i.e. basic s/w) and deliver to the customer. Then build the next part and deliver it again and repeat process.
- Iterative model it takes feedback from every phases.

# Cont'd....

- Here, we break our work into smaller parts
- Prioritize those parts and deliver to customer one by one
- Also known as “ Design a little, build a little, test a little and deploy a little model.

# Evolutionary Model



# Advantages and Disadvantages

## □ **Advantages:**

- Customer requirements are clearly specified
- Risk analysis is better
- It supports changing environment
- Initially operating time is less
- Better suited for large mission-critical projects

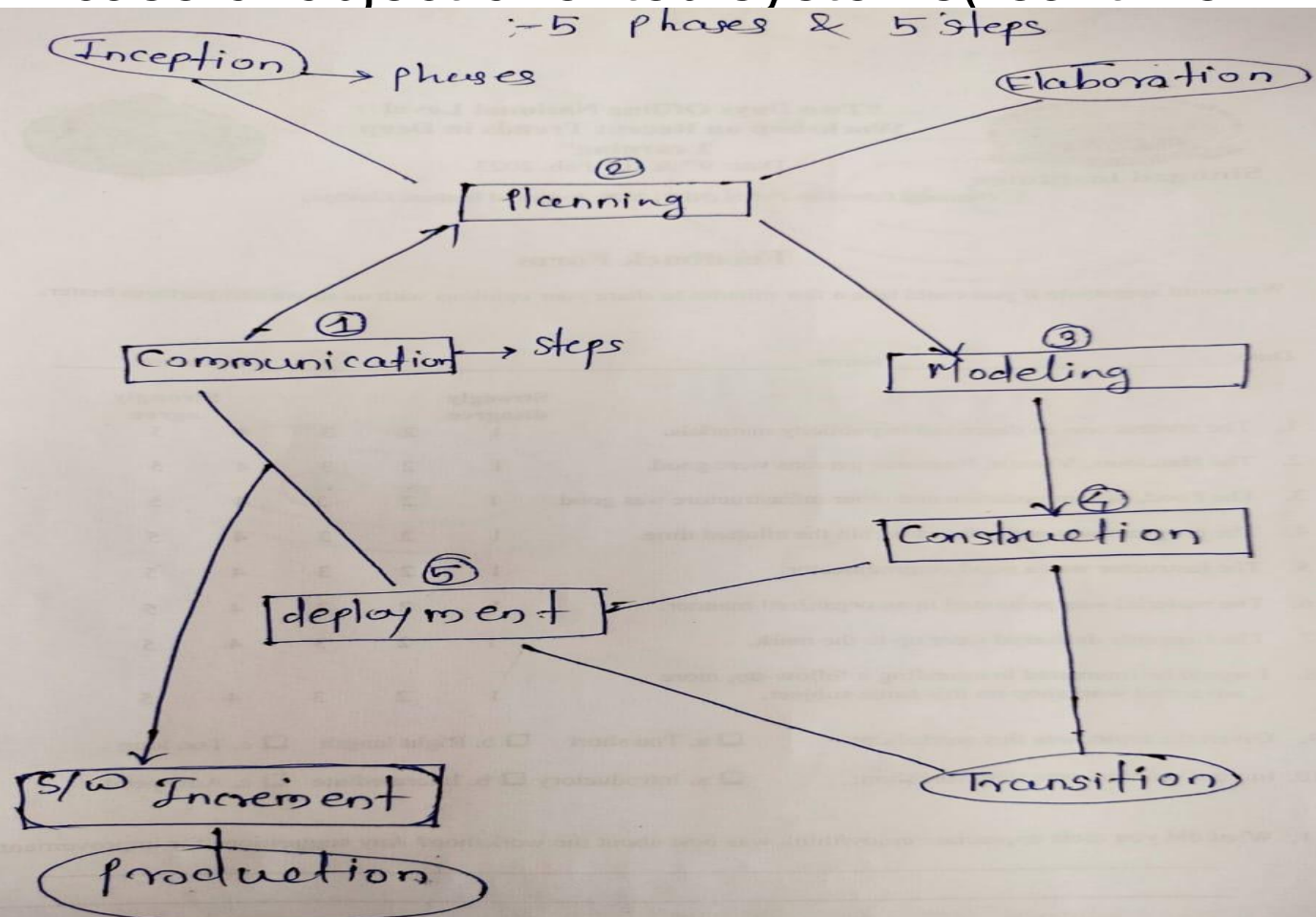
## □ **Disadvantages:**

- Not suitable for smaller projects
- Cost is large
- Highly skilled resources are required.



# Unified Process Model

- Used only in case of object oriented systems(real time systems)



# Agile Software Development

□ Agile:- Move quickly

