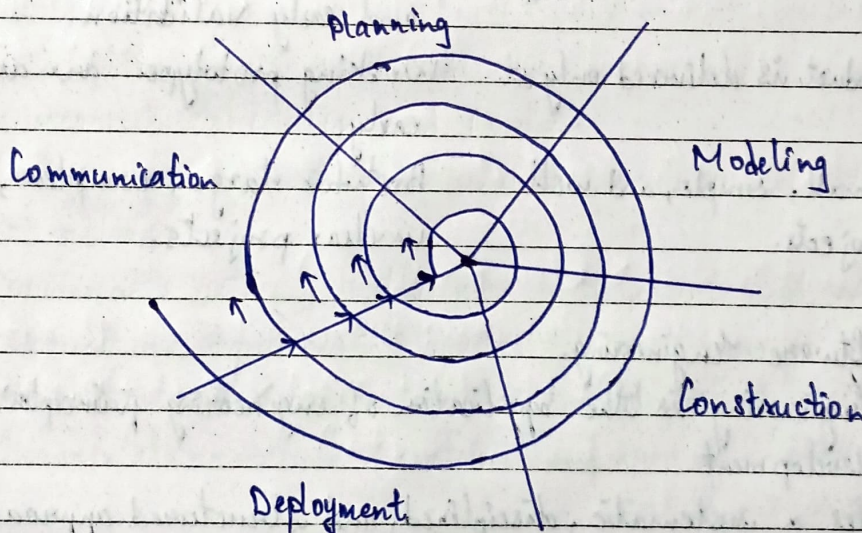


Q1] Explain spiral process model with example and diagram.

Ans-1 The spiral model was proposed by Barry Boehm in 1986.

2. It is an iterative and risk-driven software development process.
3. The model is represented as a spiral, where each loop of the spiral indicates one phase of the software.
4. The 5 phases of Spiral Model are:

- ① Communication - Collect requirements and interact with customer.
- ② Planning - Estimate resources, schedule, and project goals.
- ③ Modeling - Perform analysis and design of the system.
- ④ Construction - Coding, testing, and producing a working software build.
- ⑤ Deployment - Deliver to customer, collect feedback, and plan for next cycle.



### Spiral Model.

5. Each successive loop is a more complete and refined system.
6. Risk analysis is a key feature, making it suitable for large and complex projects.
7. Example: Consider developing an E-commerce application.  
First loop: Basic login and product catalog features.  
Second loop: Add shopping cart and secure payment gateway.  
Third loop: Add order tracking and personalized recommendations.  
Final loop: Deploy full system and gather customer feedback.



Q2] Compare Waterfall and Prototype Model.

Q4]

Ans-1.

Ans -	Waterfall Model	Prototype Model
1.	Follows a linear and sequential approach.	Follows an iterative approach based on building prototypes.
2.	Requirements must be fixed at the beginning.	Requirements are refined gradually through customer feedback.
3.	Rigid model - difficult to make changes once a phase is completed.	Flexible model - allows changes after feedback.
4.	Customer involvement is low, only at the start and end.	Customer involvement is high, throughout the development.
5.	Risks are not handled effectively.	Risks are reduced by using prototypes and early validation.
6.	Final product is delivered only at the end.	Working prototypes are available early.
7.	Best for small, simple, and well-defined projects.	Best for large, complex, or unclear projects.

Q3] Define Software Engineering.

- Ans-1.
- Software Engineering is the application of engineering principles to software development.
  - It provides a systematic, disciplined, and structured approach to build software.
  - The goal is to develop reliable, efficient, and cost-effective software systems.
  - It covers the entire software life cycle - from requirement analysis to maintenance.
  - According to IEEE, it is,  
"The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software."



Q 4] List and explain requirement engineering task.

Ans- 1. Requirement engineering is the process of defining, documenting, and maintaining software requirements.

2. The main tasks are:

① Feasibility Study.

Analyzes whether the proposed system is technically, operationally, and economically feasible.

Types: 1) Technical feasibility - checks if required technology and resources are available.

2) Operational feasibility - checks if the system can solve business problems and meet customer needs.

3) Economic feasibility - checks if the system is cost-effective and profitable.

② Requirement Elicitation and Analysis.

Collecting requirements from stakeholders (users, clients, managers).

Steps include:

1) Requirements gathering - discussion with clients and end users.

2) Organizing requirements - prioritize and arrange requirements.

3) Negotiation and discussion - resolve conflicts, clarify ambiguous requirements.

4) Documentation - record functional and non-functional requirements.

③ Software Requirement Specification.

1) Formal document prepared by the analyst after requirement collection.

2) Converts customer's informal language into technical language understandable by the development team.

3) Includes functional and non-functional requirements.

④ Requirement Validation.

1) Ensures that documented requirements are correct, complete, and feasible.

2) Checks for ambiguity, practicality, and alignment with customer needs.



3) Techniques used: Reviews/inspections, prototyping, test case generation.

### 8) Requirement Management.

- 1) Process of managing changing requirements during system development.
- 2) New requirements may emerge due to changing business needs.
- 3) Involves prioritizing requirements, handling conflicts, and adapting to technical or business environment changes.

Q5] With example explain FB Based cost estimation method.

Ans-1. The Function Point (FP) method is a software cost estimation technique that measures the functionality delivered to the user.

2. It is independent of programming language and is widely used to estimate size, effort, productivity, and cost.

3. Steps in FP Estimation:-

① Identify Information Domain Values: Count the number of: 1) External Inputs (EI), 2) External Outputs (EO), External Inquiries (EQ), Internal Logic Files (ILF), External Interface Files (EIF).

② Assign Weights: Each of the above is classified as Simple, Average, or Complex and multiplied by standard weighting factors.

③ Compute Count Total:  $\Sigma (\text{Count} \times \text{Weight})$

④ Adjust with Technical Complexity Factors: 14 General System characteristics.

⑤ Final Function Points (FP):  $FP = \text{Count-Total} \times [0.65 + 0.01 \times \Sigma(F_i)]$   
( $F_i$  i=1 to 14 are value adjustment factors)

⑥ Estimate Effort and Cost.

4. Example:

Given Data:  $EI = 24 \times 4 = 96$ ,  $EO = 46 \times 4 = 184$ ,  $EQ = 8 \times 6 = 48$ ,  $ILF = 4 \times 10 = 40$ ,  
 $EIF = 2 \times 5 = 10$ .  $\therefore$  Count Total = 378

Adjustment Factors ( $\Sigma F_i$ ) = 43, Effort = 36.9 person-months, Cost = \$7744 per month

Total pages of documentation = 387 pgs

$FP = 378 \times [0.65 + 0.01 \times 43] = 408$

Productivity =  $FP / \text{Effort} = 408 / 36.9 = 11.1 \text{ FP / PM}$

Documentation = Pages of documentation / FP =  $387 / 408 = 0.94 \text{ pages / FP}$

Cost per function = Cost / productivity =  $7744 / 11.1 = \$700 \text{ per FP}$ .

5. FP method measures software size based on functionality.