181240116001
Vishwas Acharya

SPM

Page No.: 1
Date: / /

Assignment-2

1) What is process model? Explain how to choose process model in SPM?

Ans → Different software development life cycle models have their own Pros and Cons.

→ The best model for any project can be determind by the factors like requirement, system complexity, size of the product, cost, skill limitation, etc.

- Software Process Model

  → Waterfall
  → V- Process
  → Spiral
  → Software Prototying
  → Phased development model
    → Incremental
    → Iterative

- Factors in choosing a Software Process

- Project Size : Larger projects mean bigger teams, so you'll need more extensive and elaborate project management plans

- Project Complexity : Complex projects may not have have clear requirements. The requirements may change often, and the cost of delay is high.

- Cost of delay: Is the project highly time-bound with a huge cost of delay, or are the timelines flexible?

- Customer Involvement: Do you need to consult the customers during the process? Does the user need to participate in all phases?

- Familiarity with technolog: This involves the developer's knowledge and experience with the project domain, software, tools, language, and methods needed for development.

- Project resources: This involves the amount and availability of funds, staff, and other resources.

2) Discuss the various cost benefit evaluation techniques in brief.

Ans ↳ Cost-benefit analysis (CBA) is a technique used to compare the total costs of a pramme/project with its benefits, using a common metric (mostly commonly monetary units).

↳ Project cost estimation and project scheduling are usually carried out together. The costs of development are primarily the costs of the effort involved, so the effort computation is used in both the cost and the schedule estimate.

↳ The initial cost estimates may be used to establish a budget for the project and to set a price for the software for a customer.

↳ The total cost of a software costs ~~including maintenance~~ of a software development project is the sum of following costs:

1) Hardware and software costs including maintenance.
2) Travel and training costs
3) Effort costs of paying software developers.

3) Explain COCOMO model in detail.

Ans ↳ The COCOMO (Constructive Cost Model) is one of the mostly popularly used software cost estimation models i.e. it estimates or predicts the effort required for the project, total project cost and scheduled time for the project.

↳ This model on the number of Lines of code for software product development.

↳ It was developed by a software engineer Barry Boehm in 1981.

↳ According to COCOMO, there are three modes of software development projects that depend on complexity, such as:

1.) Organic Project

↳ It belongs to small and simple software projects which are handled by a small team with good domain knowledge and few rigid requirements

2.) Semidetached Project

↳ It is an intermediate project, where the team having mixed experience to deals with rigid/nonrigid ~~do~~ requirements

3.) Embedded Project

↳ This project having a high level of complexity with a large team size by considering all sets of parameters

- Types

1.) The Basic COCOMO

↳ It is the ~~only~~ one type of static model to estimates software development effort quickely and roughly.

↳ It mainly deals with the number of lines of code and the level of estimation accuracy is less as we don't consider the all parameters belogns to the project.

$$\text{Effort (E)} = a^{*} (\text{KLOC})^{b} \text{ MM}$$

Scheduled Time (D) = $c^{*}(E)^{b}$ Months (M)

2) The Intermediate COCOMO

↳ This model estimates software development effort in terms of size of the program and other related cost drivers parameters of the project.

$$\text{Effort } (E) = a^{*}(\text{KLOC})^{b} * \text{EAF MM}$$

$$\text{Scheduled Time } (D) = c^{*}(E)^{d} \text{ Months } (M)$$

3) The Detailed COCOMO

↳ It is the advanced model that estimates the software development effort like Intermediate COCOMO in each stage of the SDLC process.

- **Advantages**

↳ Easy to estimate the total cost of the project

↳ Easy to implement with various factors

↳ Provide ideas about historical projects.

- **Disadvantages**

↳ It ignores requirements, customer skills, and hardware issues

↳ It Limits the accuracy of the software costs and mostly depends on time factors

4&5) Where Estimation Done? Differentiate top down & Bottom up estimation.

Ans 4 Top-down and bottom-up approaches are methodes used to analyze and choose securities.

-> The difference between these two estimation are given below:

TOP-Bottom (dow)	Bottom-Top (UP)
• The size of the app is estimated. The size drives algorithmic models to establish effort and schedule.	• The software components and tasks are identified. Each task is estimated. These estimates are added.
• This can be done without identifying task details. It can be orthogonal to the bottom-up approach	• Most of the steps must be performed in order to develop the software. The estimate errors tend to cancel each other.
• Early in the life cycle is may be impossible to identify tasks or comp. There may be no team in place to estimate	• Estimating software size and using the estimating models are special skills. These activities are not necessary to develop software.
• This is a natural activity for members of the agile developemet team	• This can be done by someone who is outside of the developmnt team.

6) Explain Waterfall Model and Software Prototyping Model works in SPM.

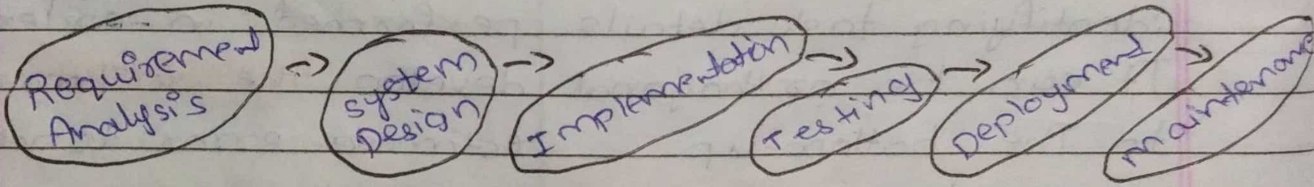
Ans: • Waterfall Model

↳ This model is the first process model to be introduced and is also known as Linear - Sequential Life cycle model

↳ In this any phase in the development process begins only if the previous phase is complete, hence the phases do not overlap.

↳ This model is divided into separate phases and typicaly, the outcome of one phase acts as the input for the next phase sequentially

• Phases



• Advantages

↳ Phases are processed and completed one at a time.

↳ Process and results are well documented

↳ Work well for smaller projects where requirements are very well understood.

- Disadvantages

↳ Not a good model for complex and object-oriented projects

↳ Poor model for long and ongoing projects.

↳ No working software is produced until late during the Life cycle

- Prototype ~~Spiral~~ model

↳ Before a design or coding proceed, a rough prototype is built to understand the requirements, and it is developed based on currenty known requirements.

↳ It is a software model and using this client can get an "actual feels" of the system, and can get better understanding of desired system.

↳ It is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determing the requirement

↳ Prototype usually not complete systems and many of the details are not buit in the prototype. The goal is to provide a system with overall functionality.

- Advantages

↳ Errors can be detected much earlier
↳ Quicker user feedback is available leading to better solutions

- Disadvantages

↳ Leads to implementing then repairing way of building system.
↳ Incomplete application may cause application not to be used as the full system was designed incomplete or inddequate problem analysis.

7) What is agile modeling? Explain Extreme Programming (XP) in brief.

Ans • Agile Model

↳ This model refers to a software developm approch based on iterative development.
↳ It breaks 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.

- Extreme Programming (XP)

↳ Lowering the cost of requirement changes in the main objective of extreme programming.

↳ XP emphasizes on final sale feedback, continuous process, shared understanding and programmer welfare

↳ In XP, there is no detailed requirements specification or software architecture built.

8) What do you understand by Incremental Delivery approach, also Explain advantages and disadvantages of using this approach.

Ans ↳ The iterative incremental model divides the product into small chunks.

↳ For Example, feature to be developed in the iteration is decided and implemented. Each iteration goes through the phases namely requirement analysis, designing, coding, and testing detailed planning is not required in iterations.

↳ Once the iteration is completed, a product is verified and is delivered to the customer for their evaluation and feedback.

↳ Hence, the product increments in terms of features and once the iterations are completed the final build holds all the features of the product.

- Phases of iterative and incremental development model:

  ↳ Inception phase
  ↳ Elaboration phase
  ↳ Construction phase
  ↳ Transition phase

- Advantages

↳ Defects are detected at an early stage
↳ Risk is analyzed and identified in the iterations.
↳ As the product is divided into smaller chunks it is easy to manage the product.

- Disadvantages

↳ Complete requirement and understanding of a product are required to break down and build incrementally.

Q. State the characteristics of the "V shape SDLC model".

Ans ↳ V shape model is also known as verification and Validation model.
↳ Here, development and testing goes parallel.
↳ Hence, V model and waterfall model are same.

↳ There are two phases, which are
a) Verification
b) Validation

a) Verification

(i) Requirement analysis : Here, required info. is gathered and analyzed.

(ii) System design : architecture, components of the product are created and documented in a design document.

(iii) High-level design : It defines the functionalities between two modules.

(iv) Low-level design : It defines the architecture or design of individual components.

(v) Coding : Code development is done in this phase.

b) Validation

(i) Unit testing : It is performed on individual components which lead to early defect detection.

(ii) Integration testing : It is done on the integrated modules and performed by testers

(iii) System testing : The complete system is tested i.e. the entire system functionality is tested.

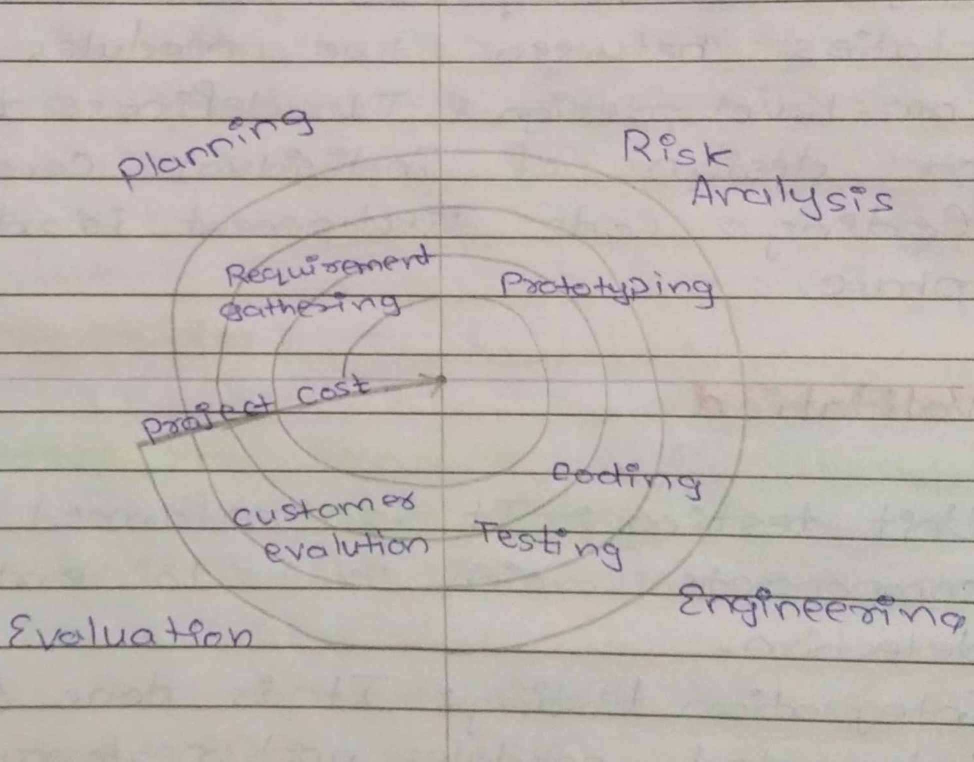
(iv) Acceptance testing : It is associated with requirement analysis phase and is done in the customer's environment.

10) Explain Spiral Model in brief.

Ans ↳ This model is an evolutionary software process model that couples that iterative nature of prototyping with controlled and systematic aspects of waterfall model.

↳ It provides potential for rapid devopment of increasingly technology.

↳ It has four phases : Planning, Risk Analysis, Engineering and Evaluation



- Advantages

↳ Software is produced early in the software life cycle.

↳ Strong approval and documentation control.

↳ Additional functionality can be added to a later date.

- Disadvantages

↳ Can be costly model to use
↳ Doesn't work well for smaller Profect
↳ Profect's success is highly dependent on the risk analysis phase.