

## Question Bank

### 1. What is software

Ans) pg 1-4

- \* what is software
- Number of programs written or made to complete ~~on~~ the given particular tasks.
- When we make a software for someone → product it becomes
  
- \* Types of software
- Generic - For all (Eg: FB / whatapp)
- Customised - For specific (Eg: Bank).
  
- \* Characteristics of software (5 marks)
- Software are engineered and not manufactured
- Software does not ~~wear~~ out (never gets kharab)
- ie only gets updated
- Software is customer ~~or show~~ build.  
(Eg: what to give to teacher)

### 2. What is software engineering

Ans) pg 1-7

No. of programs written or made to complete a given particular task is known as Software. A systematic way to build software is known as Software Engineering.

### 3. Changing nature of the software

Ans) pg 1-6

- \* Changing nature of software
- Engineering software (can be changed)
- Embedded software (need only memory stored Eg: oven) (Software install in read only memory)
- Web applications (need internet)
- AI applications.

### ► 3. Engineering/scientific software

- These types of software are characterized through the “number crunching” algorithms. There are number of engineering software in various fields like astronomy, space shuttle orbital dynamics, automotive stress analysis, automated manufacturing, molecular biology etc.
- However, there are drastic changes in modern applications within the engineering/scientific area as they are shifting from traditional numerical algorithms.

### ► 4. Embedded software

These are the software which are merged within electronic devices and are used for the purpose of implementing and controlling the features as well as functions for the end user as well as system itself.

### ► 6. Web applications

- These applications are also called as “WebApps”. This is a network-centric software category which covers large number of applications.
- In shortest form, WebApps are same as of the group of linked hypertext files which provide information through text and limited graphics.
- Now-a-days the modern WebApps are evolving into environments of sophisticated computing which offer stand-alone features, computing functions, as well as content to the end user.
- It is also possible to integrate WebApps with corporate databases and business applications.

### ► 7. Artificial intelligence software

- In these applications non-numerical algorithms are generally used for the purpose of solving complex problems which cannot be handled by computation or straightforward analysis.
- There are various applications in this category such as robotics, games, expert systems, artificial neural networks, proving theorem, pattern matching like image and voice.

#### **4. What is the use of software process?**

Ans) Software process refers to a set of activities and steps that are followed to develop, design, test, deploy and maintain software products. The primary use of software process is to provide a structured and systematic approach to software development that helps ensure high-quality software products that meet user requirements and are delivered on time and within budget.

The following are some of the key uses of using software process:(RIICS)

1. **Consistency:** By following a set of established processes, software development teams can ensure that consistent and repeatable methods are used throughout the development lifecycle. This reduces the likelihood of errors and inconsistencies that can lead to poor quality or unreliable software.
2. **Risk management:** Software development projects can be complex and risky, involving many unknowns and potential pitfalls. A well-defined software process helps to identify and manage risks throughout the development lifecycle, reducing the likelihood of project failures or delays.
3. **Improved quality:** The use of a software process ensures that quality is built into the development process from the start. This includes activities such as requirements gathering, design, coding, testing, and deployment, which are all designed to ensure that the software meets user requirements and is free of defects.
4. **Increased productivity:** A well-defined software process can help to streamline development activities, reducing the time and effort required to develop and deploy software. This can result in increased productivity and cost savings for software development teams.
5. **Scalability:** Software processes can be adapted to suit the needs of different projects, teams, and organizations. This means that they can be scaled up or down as needed to accommodate changing requirements, team sizes, and project complexities.

## 5. Explain CMM

Ans) pg 1-10, 1-11

(Introduction to Software Engineering)...Page no. (1-10)

Software Engg. & Project Mgmt (MU-Sem 6-AIDS)

ideas which are proved. Software engineering, in contrast, makes use of limited, but global concepts such as loops, conditions etc.

### 1.5 CAPABILITY MATURITY MODEL (CMM)

**UQ.** Explain CMM Level 4 and 5. MU - Dec. 15, 5 Marks

**UQ.** What are the different levels of Capability Maturity Model (CMM) ? MU - May 16, Dec 17, May 18, 5 Marks

**UQ.** Explain CMM. MU - Dec. 16, Dec 17, 5 Marks

- The Capability Maturity Model (CMM) invented by Software Engineering Institute (SEI) states a growing sequence of levels of a software development organization.
- The higher the level, the better the software development procedure so reaching every level is a cost consuming and time consuming procedure.
- The Capability Maturity Model (CMM) is a mechanism used to create and refine software creation procedure of organization.
- The CMM is same as ISO 9001, one of the ISO 9000 sequences of standards determined by the International Organization for Standardization (ISO).
- The ISO 9000 standards state an efficient quality system for manufacturing and service industries; ISO 9001 applied particularly to software development and maintenance.
- The major variation among the two systems present in their uses : ISO 9001 states a lowest acceptable quality level for software procedures, while the CMM use a framework for uninterrupted procedure improvement and is more explicit than the ISO standard in defining

Following are levels of CMM :

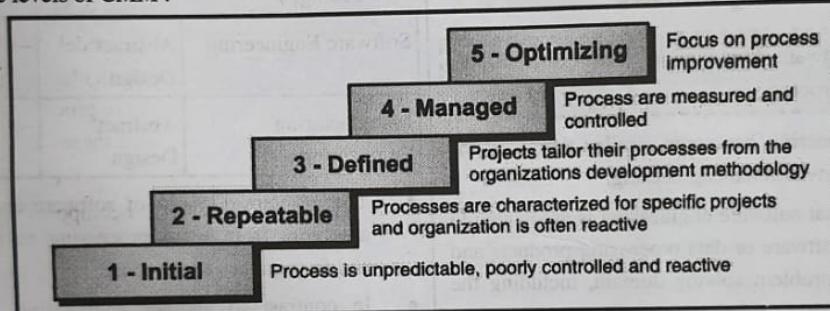


Fig. 1.5.1 : CMM Levels

Q Explain SEI CMM model? → exam

PAGE No	/ /
DATE	

**CMM - Capability Maturity Model.**

- ⇒ Developed by → SEI (Software Engineering Institute)
- SEI : ⇒ CMM → to know the level of different process in industry.
- Has five parameters knowing levels to measure

level 1 : Initial

level 2 : Repeatable → S/W Project planning, Requirement management, etc.

level 3 : Defined → Peer review, Training Programs, (using proper method) Unchangeable Integrated S/W management, etc.

level 4 : Managed → S/W quality management, Quantitative (maintain quality) Process management

level 5 : Optimizing → Process change management, (implement new things) technology change, Defect prevention.

(7) Agile Model (It is a methodology) → SRS is generated as an iterative development

**Initial** – Work is performed informally i.e organization is not planned in advance.

**Repeatable** – Work is planned and tracked.

This level of software development organization has basic and consistent project management processes to track cost, schedule & time functionality.

**Defined** – Work is well defined.

At this level the software process for both management and engineering activities are defined and documented.

**Managed** – Work is quantitatively & qualitatively controlled.

**Optimizing** – Work is Based Upon Continuous Improvement.

## **6. What is SDLC(SOFTWARE DEV LIFECYCLE)**

Ans) A life cycle model represents all the methods required to make a software product transit through its life cycle stages. It also captures the structure in which these methods are to be undertaken.

### **1. Preliminary Investigation:**

It is an initial phase at the start of a project that determines whether the concept is viable; any proposed alternative solutions; the cost benefit evaluation; and finally the submission of the preliminary plan for recommendations, i.e here the research and development regarding the existing project takes place.

### **2. Requirement Gathering:**

Gathering requirements for the project is the most important part of the SDLC for project managers and internal stakeholders of a project. During this phase, the customer states the expectations of the project including who will use the product, how the customer will use the product, and the specific information included with any special customer requirements related to the software. Once the requirement gathering is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document. A feasibility study is also carried out in this phase. A feasibility analysis assesses the project's likelihood of success. Types of feasibility studies include Financial Feasibility, Technical Feasibility, Economic Feasibility, Man power, etc.

### **3. Design:**

In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived. The project supervisors use suitable designs to choose the definitive design that you can produce for the project.

### **4. Construction:**

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per design during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle. The programming language is chosen with respect to the type of software being developed.

## **5. Testing:**

This stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

## **6. Deployment:**

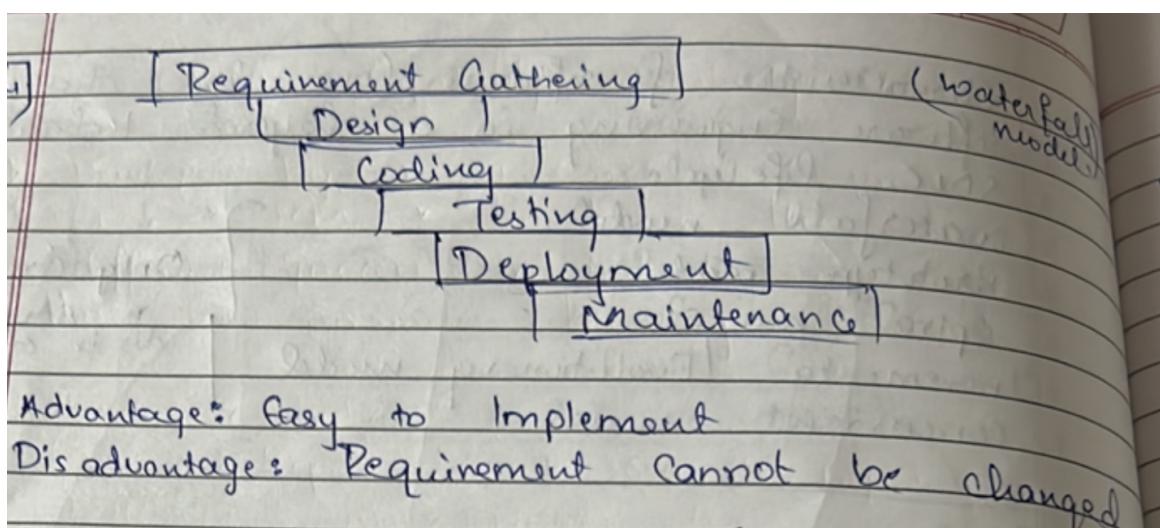
Once the product is tested and ready to be deployed it is released formally in the appropriate market. Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment.

**7. Maintenance:** Once when the client starts using the developed systems, then the real issues come up and requirements are to be solved from time to time. This procedure where care is taken for the developed product is known as maintenance. (handle bugs and errors)

## **7. Write SN on the following topics: (write in detail draw diag.)**

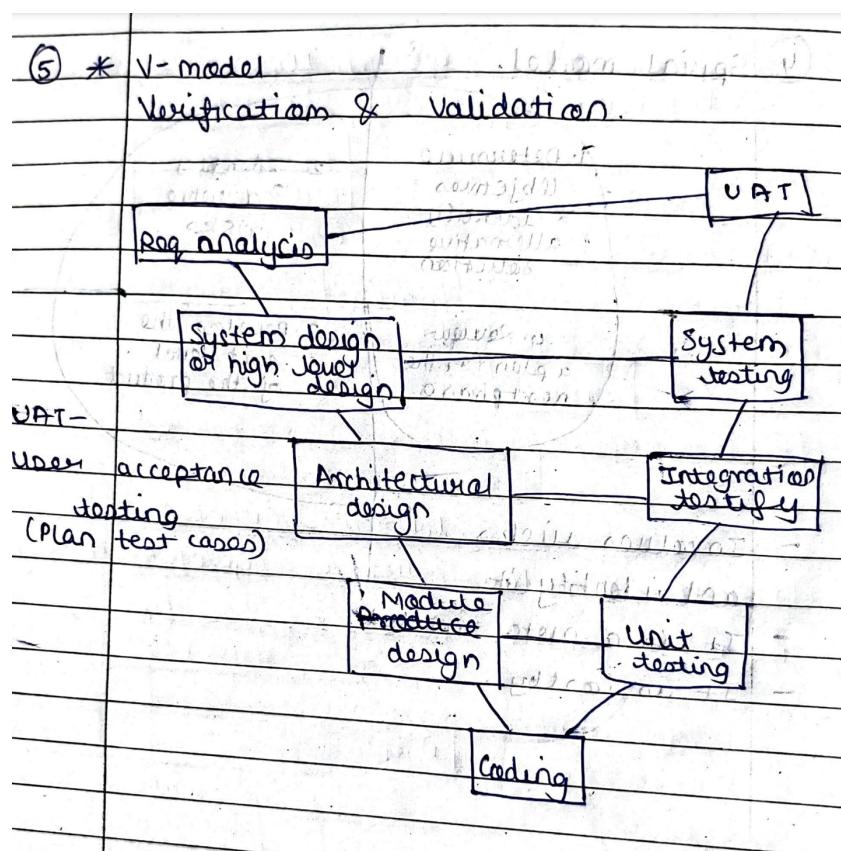
- waterfall model
- V model
- Incremental process model
- Prototype model
- Spiral model

Ans) - Waterfall model



- Waterfall model is the first approach used in software development process.
- It is also called as classical life cycle model or linear sequential model.
- In waterfall model any phase of development process begins only if previous phase is completed.

### - V Model



The V-model is a type of SDLC model where the process executes in a sequential manner in V-shape. It is also known as the Verification and Validation model. It is based on the association of a testing phase for each corresponding development stage. Development of each step directly associated with the testing phase.

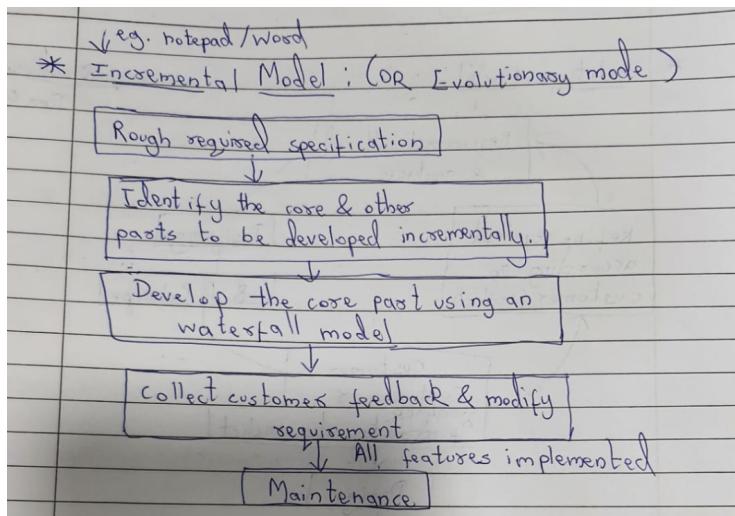
### **Advantages:**

1. Easy to Understand.
2. Testing Methods like planning, test designing happens well before coding.
3. This saves a lot of time. Hence a higher chance of success over the waterfall model.

### **Disadvantages:**

1. Very rigid and least flexible.
2. Not good for a complex project.

#### **- Incremental Process Model**

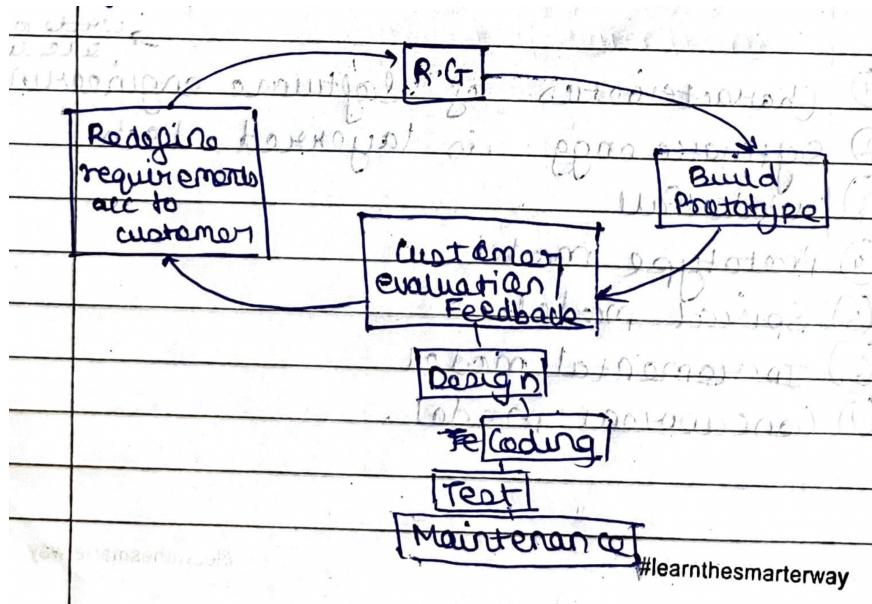


- The incremental model applies the waterfall model incrementally.
- The series of releases is referred to as "increments", with each increment providing more functionality to the customers.

**Advantage:** It is less time-consuming.

**Disadvantage:** It is more complex.

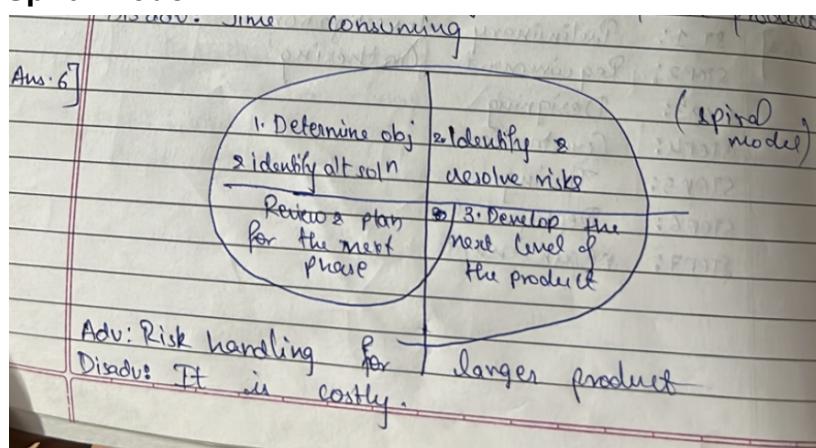
#### **- Prototype Model:**



- Prototyping allows user to evaluate developers' proposal and try it before actual implementation.
- Prototyping model is widely used popular software development model as it helps to understand user requirements in early stage of development process.

Adv: Less chance to generate defective product  
 Disadv: Time consuming

### - Spiral Model



- Spiral model is a combination of iterative model and waterfall model.
- Spiral model has four phases of development each of these phases is called as spiral.

## 8. Explain agile methodology in detail. (list methods and explain in short)

Ans)

- Agile software development describes an approach to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customers (end users).

- It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages rapid and flexible response to change.
- There is significant subjective evidence that adopting agile practices and values improves the agility of software professionals, teams and organizations.

### \* Types of Agile methodology:

1. Scrum
2. Extreme programming
3. Crystal

#### 1. Scrum:

Scrum is an agile framework for managing knowledge work, with an emphasis on software development. It is designed for teams of three to nine members, who break their work into actions that can be completed within time boxed iterations, called "sprints", no longer than one month and most commonly two weeks, then track progress and re-plan in 15-minute stand-up meetings, called daily scrums.

Scrum principles are mostly based on the agile manifesto and help to guide activities regarding development inside a process which includes framework activities such as requirements, analysis, design, evolution, and delivery.

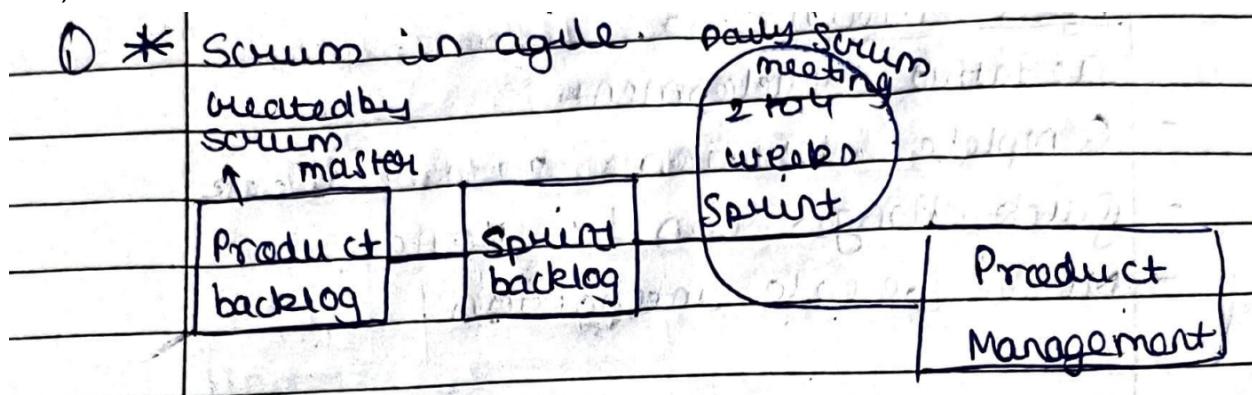
In all the framework activities, work tasks happen within a process pattern called a sprint.

## 2. Extreme Programming:

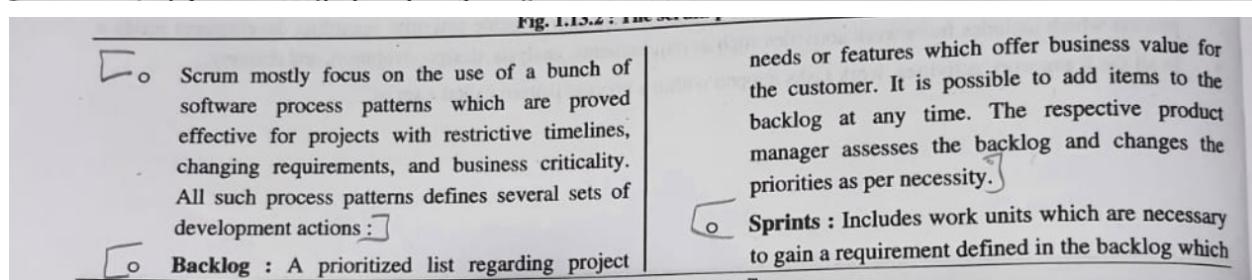
- Extreme Programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements.
- As a type of agile software development, it advocates frequent "releases" in short development cycles, which is intended to improve productivity and introduce checkpoints at which new customer requirements can be adopted.

## 9. Write an SN on SCRUM (whole scrum process with diag)

Ans)



Scrum is an agile framework for managing knowledge work, with an emphasis on software development. It is designed for teams of three to nine members, who break their work into actions that can be completed within time boxed iterations, called "sprints", no longer than one month and most commonly two weeks, then track progress and re-plan in 15-minute stand-up meetings, called daily scrums.

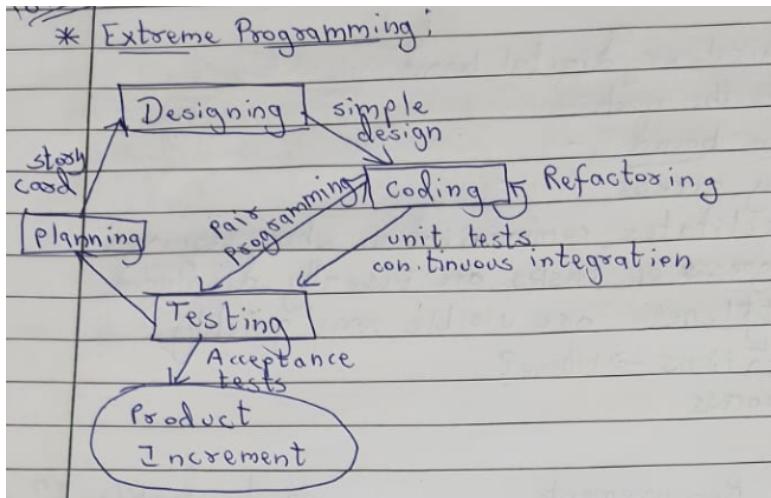


- (MO-Sem 6-AIDS)
- should be incorporated in predefined time-box (usually thirty days).
- o Modifications (such as backlog work items) are not involved during the sprint. Therefore, team members are allowed by the sprint to work in a short-term, but stable environment.
  - o Scrum meetings - are short (usually fifteen minutes) meetings conducted on daily basis by the Scrum team. Three most important key questions which are asked and must be answered during the meeting are:
    - What has been done since the last meeting?
    - What will be done in the next sprint?
    - What impediments are there?
- encourages a strong team structure.
- o Demos : Implement the software increment at the client side (customer) so that the newly developed functionality can be demonstrated as well as evaluated by the customer.
  - o It is also possible that the demo may not able to contain all planned functionality, but instead such functions which can be delivered within the established time-box.

## 10. Explain extreme programming

Ans)

- Extreme Programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements.
- As a type of agile software development, it advocates frequent “releases” in short development cycles, which is intended to improve productivity and introduce checkpoints at which new customer requirements can be adopted.



- 5 values of XP
- ① Communication
  - ② Simplicity
  - ③ Feedback
  - ④ Courage
  - ⑤ Respect

(2CFRS)

### Communication

XP stresses the importance of the appropriate kind of communication – face to face discussion with the aid of a white board or other drawing mechanism.

### Simplicity

The purpose of this is to avoid waste and do only absolutely necessary things such as keep the design of the system as simple as possible so that it is easier to maintain, support, and revise.

### Feedback

Through constant feedback about their previous efforts, teams can identify areas for improvement and revise their practices. Feedback also supports simple design.

## Courage

You need courage to raise organizational issues that reduce your team's effectiveness, to stop doing something that doesn't work and try something else, to accept and act on feedback, even when it's difficult to accept.

## Respect

The members of your team need to respect each other in order to communicate with each other, provide and accept feedback to work together and identify simple designs and solutions.

### 11. Explain Kanban model

Ans)

- Kanban is a popular framework used to implement agile software development. It requires real-time communication of capacity and full transparency of work.
- Work items are represented visually on a kanban board, allowing team members to see the state of every piece of work at any time.
  
- The work of all kanban teams revolves around a kanban board, a tool used to visualize work and optimize the flow of the work among the team.
- While physical boards are popular among some teams, virtual boards are a crucial feature in any agile software development tool for their traceability, easier collaboration, and accessibility from multiple locations.
- Regardless of whether a team's board is physical or digital, their function is to ensure the team's work is visualized, their workflow is standardized, and all blockers and dependencies are immediately identified and resolved.
- A basic kanban board has a three-step workflow: To Do, In Progress, and Done. However, depending on a team's size, structure, and objectives, the workflow can be mapped to meet the unique process of any particular team.

Kanban cards

To do	In development	In testing	Ready for Release	Released
Task 1 Priority 2	Task 9 Owner RV	Task 15 Owner PD	Task 6	Task 50
Task 2 Priority 3	Task 11 Owner SS	Task 10 Owner BS	Task 9	Task 20
Task 3 Priority 1	Task 20 Owner SS	Task 8 Owner PM	Task 25	Task 30

(P, T, O)

## 12. What is the requirement explain its types

Ans)

GQ. What do you mean by requirement? (2 Marks)

- Requirement the information which describes the user's expectation about the system performance is called as requirement.
- The requirement must be clear and unambiguous. Some requirement definition has to face problem of lack of clarity.

### Types of requirements

- Functional requirements
- Non-functional requirements
- Domain requirements

Functional:

- In software engineering, a functional requirement is used to define a function regarding a system or its component, in which a function is described as a specification of behavior between outputs and inputs.

### Non-Functional:

- These are basically the quality constraints that the system must satisfy according to the project contract.
- The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.
- They basically deal with issues like :
  - Portability
  - Security
  - Maintainability
  - Reliability
  - Scalability
  - Performance
  - Reusability
  - Flexibility

### Domain:

- Domain requirements are used to describe system characteristics as well as features which impact the domain. Those may be new functional requirements, constraints on present requirements or may define particular computations.
- If domain requirements are not satisfied, the system may be unworkable.
- **Example :** Library system

## 13. Explain the requirement engineering process in detail

Ans)

- The procedure which collects the software requirements from customer, analyze and document them is called as **requirement engineering**.
- The aim of requirement engineering is to create and maintain ‘System Requirements Specification’ document.

The process Requirement Engineering includes are as follows:

1. Requirement elicitation :- Interviews, survey, observations, focus group, (or gathering.)
2. Requirement analysis.
3. Requirement specification:- Prioritizing
4. Requirement verification & validation:- Review requirements document holding meeting
5. Requirement management:- Tracking & Controlling changes.
  - Version control
  - Traceability.
  - Communication.

#earnthesmarterway  
#reporting

### Requirement Elicitation:

In requirements engineering, requirements elicitation is the practice of researching and discovering the requirements of a system from users, customers, and other stakeholders. The practice is also sometimes referred to as "requirement gathering".

### Requirement Analysis:

Requirements are identified (including new ones if the development is iterative) and conflicts with stakeholders are solved. Both written and graphical tools (the latter commonly used in the design phase but some find them helpful at this stage, too) are successfully used as aids Examples of written analysis

### **Requirement Specification:**

- Requirements are documented in a formal artifact called Requirements Specification (RS).
- Nevertheless, it will become official only after validation. A RS can contain both written and graphical (models) information if necessary. Example : Software Requirements Specification (SRS).

### **Requirement Verification and Validation:**

#### **6. Requirements**

It is the process of checking whether the documented requirements and models are consistent and meet the

needs of the stakeholder. Only if the final draft passes the validation process, the RS becomes official.

### **Requirement Management:**

Managing all the activities related to the requirements since inception, supervising as the system is developed and, even until after it is put into use (e. g., changes, extensions, etc.)

### **14. What is 3 P's**

Ans)

The key to increasing the chances of success on software projects is to maintain the proper relationships among people, process, and product -- the three "Ps."

### People

- The primary element of any project is the people. People gather requirements, people interview users (people), people design software, and people write software for people. No people -- no software.

### Process

- Process is how we go from the beginning to the end of a project. All projects use a process. Many project managers, however, do not choose a process based on the people and product at hand. They simply use the same process they've always used or misused.

building it.

### Product

- The product is the result of a project. The desired product satisfies the customers and keeps them coming back for more. Sometimes, however, the actual product is something less.
- The product pays the bills and ultimately allows people to work together in a process and build software.

Always keep the product in focus.

## 15. What is SRS

Ans)

GQ. What is SRS document ?

(4 Marks)

- SRS stands for Software/System Requirement Specification. SRS is a special kind of document which contains user requirements for a system which states properties and constraints that must be satisfied by a software system.
- IEEE defines software requirements specification as, 'a document that clearly and precisely describes each of the essential requirements (functions, performance, design constraints and quality attributes) of the software and the external interfaces.'
- The outcome of the requirements gathering and analysis phase of the SDLC is **Software Requirements Specification (SRS)**.
- SRS is also called as **requirements document**.
- SRS is base for software engineering actions and is generated when all requirements of software project are gathered and analyzed.
- It is generally signed at the end of requirements engineering phase.

## **16. What is a process and project metrics.**

Ans) A software metric is a measure of software characteristics which are measurable or countable. Software metrics are valuable for many reasons, including measuring software performance, planning work items, measuring productivity, and many other uses.

**Process Metrics:** These are the measures of various characteristics of the software development process. For example, the efficiency of fault detection. They are used to measure the characteristics of methods, techniques, and tools that are used for developing software.

**Project Metrics:** Project metrics are the metrics used by the project manager to check the project's progress. Data from the past projects are used to collect various metrics, like time and cost; these estimates are used as a base of new software. As the project proceeds, the project manager will check its progress from time-to-time and will compare the effort, cost, and time with the original effort, cost and time.

## **17. Explain project estimation techniques (LOC AND FP methods or sums)**

Ans) Estimation of the size of the software is an essential part of Software Project Management. It helps the project manager to further predict the effort and time which will be needed to build the project. Various measures are used in project size estimation. Some of these are:

- Lines of Code
- Function points, etc.

### **1. Lines of Code (LOC):**

1. Line of code (LOC) is a software metric used to measure the size of a computer program by counting the number of lines in the text of the program's source code.
2. It is one of the earliest and simpler metrics for calculating the size of the computer program. It is generally used in calculating and comparing the productivity of programmers.

### **2. Functional Point Analysis(FP):**

1. The basic and primary purpose of the functional point analysis is to measure and provide the software application functional size to the client, customer, and the stakeholder on their request.
2. The functional size of the product is measured in terms of the function point, which is a standard measurement to measure the software application.

FP is calculated by following formula:

$$FP = \text{Count-total} * [0.65 + 0.01 * \sum(f_i)]$$

$$CAF = [0.65 + 0.01 * \sum(f_i)]$$

= Count-total \* CAF

where Count-total is obtained from the Table given & CAF is Complexity Adjustment Factor and  $\sum(f_i)$  is the sum of all 14 questionnaires.

### 18. Explain the cocomo model

Ans) COCOMO is one of the most generally used software estimation models in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software.

The necessary steps in this model are:

1. Get an initial estimate of the development effort from evaluation of thousands of delivered lines of source code (KLOC).
2. Determine a set of 15 multiplying factors from various attributes of the project.
3. Calculate the effort estimate by multiplying the initial estimate with all the multiplying factors i.e., multiply the values in step1 and step2.

In COCOMO, projects are categorized into three types:

1. Organic
2. Semi Detached
3. Embedded

**1.Organic:** A development project can be treated of the organic type, if the project deals with developing a well-understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar methods of projects.

**2. Semi Detached:** A development project can be treated with semi detached type if the development consists of a mixture of experienced and inexperienced staff. Team members may have finite experience in related systems but may be unfamiliar with some aspects of the order being developed.

**3. Embedded:** A development project is treated to be of an embedded type, if the software being developed is strongly coupled to complex hardware, or if the stringent regulations on the operational method exist. For Example: ATM, Air Traffic control.

Software cost estimation should be done through three stages:

1. Basic Model - provides an accurate size of project parameter.
2. Intermediate Model - recognizes facts & refines initial estimates.
3. Detailed Model - incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step.

## **19. What is design process & model**

Ans) Software design is a mechanism to transform user requirements into some suitable form, which helps the programmer in software coding and implementation. The main aim of design engineering is to generate a model which shows firmness, delight and commodity. Software design is an iterative process through which requirements are translated into the blueprint for building the software.

Objectives:

1. **Correctness:** Software design should be correct as per requirement.
2. **Completeness:** The design should have all components like data structures, modules, and external interfaces, etc.
3. **Efficiency:** Resources should be used efficiently by the program.
4. **Flexibility:** Able to modify on changing needs.
5. **Consistency:** There should not be any inconsistency in the design.
6. **Maintainability:** The design should be so simple so that it can be easily maintainable by other designers.