SEN QUESTION BANK SOLVE

1. Define software? Draw the failure curve for software.

In a computer system, the software is basically a set of instructions or commands that tells a computer what to do. Or in other words, the software is a computer program that provides a set of instructions to execute a user’s commands and tell the computer what to do

2. explain software characteristics?

* [Top Characteristics of Software](https://www.interviewbit.com/blog/what-are-the-characteristics-of-software/" \l "top-characteristics-of-software)
  + [Functionality](https://www.interviewbit.com/blog/what-are-the-characteristics-of-software/" \l "functionality)
  + [Usability (User-friendly)](https://www.interviewbit.com/blog/what-are-the-characteristics-of-software/" \l "usability-user-friendly)
  + [Efficiency](https://www.interviewbit.com/blog/what-are-the-characteristics-of-software/" \l "efficiency)
  + [Flexibility](https://www.interviewbit.com/blog/what-are-the-characteristics-of-software/" \l "flexibility-)
  + [Reliability](https://www.interviewbit.com/blog/what-are-the-characteristics-of-software/" \l "reliability)
  + [Maintainability](https://www.interviewbit.com/blog/what-are-the-characteristics-of-software/" \l "maintainability)
  + [Portability](https://www.interviewbit.com/blog/what-are-the-characteristics-of-software/" \l "portability)
  + [Integrity](https://www.interviewbit.com/blog/what-are-the-characteristics-of-software/" \l "integrity)
* [Conclusion](https://www.interviewbit.com/blog/what-are-the-characteristics-of-software/" \l "conclusion)
* [Additional Resources](https://www.interviewbit.com/blog/what-are-the-characteristics-of-software/" \l "additional-resources)

3. Types of software?

* Driver software. ...
* Middleware. ...
* Programming software
* System software. ...
* Utility software. ...
* Application software.
* Embedded software.

4. Explain the layered approach of software engineering?

[Software engineering](https://www.geeksforgeeks.org/software-engineering-introduction-to-software-engineering/)is a fully layered technology, to develop software we need to go from one layer to another. All the layers are connected and each layer demands the fulfillment of the previous layer.

* + 1. **A quality focus:**It defines the continuous process improvement principles of software.
    2. **Process:**It is the foundation or base layer of software engineering. It is key that binds all the layers together which enables the development of software before the deadline or on time.
    3. **Method:**During the process of software development the answers to all “how-to-do” questions are given by method. It has the information of all the tasks which includes communication, requirement analysis, design modeling, program construction, testing, and support.
    4. **Tools:**Software engineering tools provide a self-operating system for processes and methods. Tools are integrated which means information created by one tool can be used by another.

5. Explain waterfall process models?

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

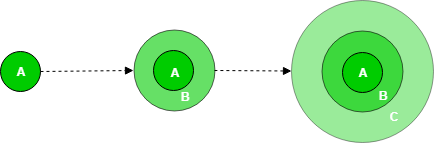
The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.



6. Explain the Incremental process model?

The incremental process model is also known as the Successive version model.

First, a simple working system implementing only a few basic features is built and then that is delivered to the customer. Then thereafter many successive iterations/ versions are implemented and delivered to the customer until the desired system is released.



7. Explain the Spiral Model?

8. Explain the RAD Model? write its Advantages and disadvantage?

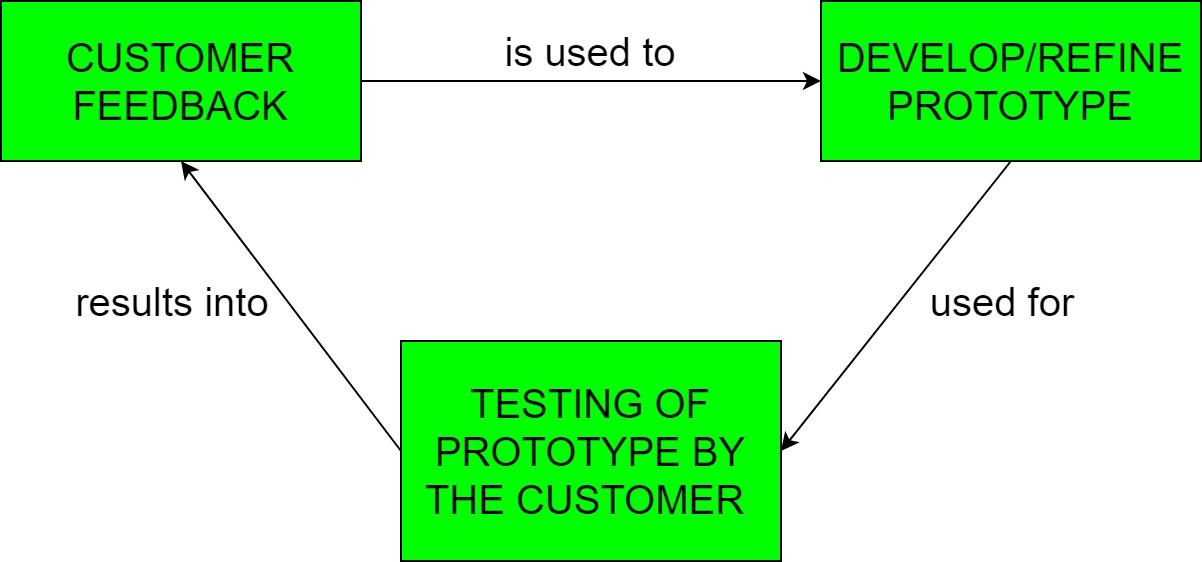
The Rapid Application Development (or RAD) model is based on prototyping and iterative model with no (or less) specific planning. In general, RAD approach to software development means putting lesser emphasis on planning tasks and more emphasis on development and coming up with a prototype.

Emphasis=( (giving) special importance or attention (to something));

| **Advantages of RAD Model** |
| --- |
| * Flexible and adaptable to changes |
| * It is useful when you have to reduce the overall project risk |
| * It is adaptable and flexible to change |
| * With less people, productivity can be increased in short time  | **Disadvantages of RAD Model** | | --- | | * It can’t be used for smaller projects | | * Not all application is compatible with RAD | | * When technical risk is high, it is not suitable * Requires highly skilled designers or developers | |

9. Explain the prototype Model?

The Prototyping Model is one of the most popularly used Software Development Life Cycle Models (SDLC models). This model is used when the customers do not know the exact project requirements beforehand. In this model, a prototype of the end product is first developed, tested and refined as per customer feedback repeatedly till a final acceptable prototype is achieved which forms the basis for developing the final product.



10. Explain the Extreme programming Model?

Extreme programming (XP) is one of the most important software development frameworks of Agile models. It is used to improve software quality and responsiveness to customer requirements. The extreme programming model recommends taking the best practices. **Good practices need to be practiced in extreme programming.** Some of the good practices that have been recognized in the extreme programming model are as follow-

* **Code Review**
* **Testing**
* **Simplicity**
* **Design**
* **Listening**
* **Feedback**

11. Define the Umbrella Activity of the process framework?

Umbrella activities are a set of steps or procedure that the software engineering team follows to maintain the progress, quality, change and risks of the overall development tasks.

**Umbrella Activities are as follows:**

* Software Project Tracking and Control.
* Formal Technical Reviews.
* Software Quality Assurance.
* Software Configuration Management.
* Document Preparation and Production.
* Re-usability Management.
* Measurement and Metrics.
* Risk Management.

12. Prescriptive process model and agile process model.

* The **prescriptive process** models stress detailed definition, identification, and application of process activates and tasks. Intent is to improve system quality, make projects more manageable, make delivery dates and costs more predictable, and guide teams of software engineers as they perform the work required to build a system.

Unfortunately, there have been times when these objectives were not achieved. If prescriptive models are applied dogmatically and without adaptation, they can increase the level of bureaucracy.

* **Agile process models** emphasize project “agility” and follow a set of principles that lead to a more informal approach to software process. It emphasizes maneuverability and adaptability. It is particularly useful when Web applications are engineered.

13. what is the Agile process model? Write Advantages and disadvantages.

**Agile development model** is also a type of **[Incremental model](https://tryqa.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/" \o "What is Incremental model- advantages, disadvantages and when to use it?)**. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly **[tested](https://tryqa.com/why-is-testing-necessary/" \o "Why is testing necessary?)** to ensure **[software quality](https://tryqa.com/what-is-software-quality/" \o "What is Software Quality?)** is maintained.

**Advantages of Agile model:**

* Customer satisfaction by rapid, continuous delivery of useful software.
* People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
* Working software is delivered frequently (weeks rather than months).
* Face-to-face conversation is the best form of communication.
* Close, daily cooperation between business people and developers.
* Continuous attention to technical excellence and good design.
* Regular adaptation to changing circumstances.
* Even late changes in requirements are welcomed

**Disadvantages of Agile model:**

* In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
* There is lack of emphasis on necessary designing and documentation.
* The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
* Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

14. List and explain the “Core Principle of function of requirement engineering processes Explain six function of requirement engineering processes” of Software Engineering?

15. what is scrum?

Scrum is an agile project management framework that helps teams structure and manage their work through a set of values, principles, and practices. Much like a rugby team (where it gets its name) training for the big game, scrum encourages teams to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses to continuously improve

Unit 2

1. **write the importance of Software Engineering Practices?**

The software process provides everyone involved in the creation of a computer-based system with a road map for getting to successful destination.

Practice provides you with the details you will need to drive along the road.

Practice helps you to understand the concepts and principles that must be understood and followed to drive safely and rapidly.

Practice instructs you on how to drive, where to slow down and where to speed up.

1. **write the core principle of software engineering?**

Principle #1: The reason it all exists: A software system exists for one important reason, to provide value to its users.

Principle #2 Keep it simple,Stupid(KISS):There are many terms and factors to consider in any software design effort.

Principle #3 Maintain the vision:A clear vision is essential and important to the success of a software project software without one, a project almost unfailingly ends up being, of two or more minds about itself.

Principle #4 What you produce, other will consume: Always specify, design and implement knowing someone else will have to understand what you are doing.

1. **Describe principles of communication for software engineering?**

Principle #1 Listen carefully: To understand customer’s requirements carefully and perfectly. Listening principle ensures the proper data collection from the speaker. You should ask questions for doubts mind if you have, and get them clarifies. Try to avoid interruption in between.There should not be arguments in talk.

Principle #2 Preparation before communication: It is very important to prepare the agenda for the meeting or discussion. The agenda contains points to be discussed in the meeting.

Principle #3 Be prepared facilitate the activity: The successful communication meeting which it was held should have a leader who keeps the conversation moving in a productive direction.

Principle #4:Face-to-face communication: Face-to-face communication is better then communicating from a far distance it is even better if a document related to that particular discussion is given.

1. **Describe principles of planning for software engineering?**

Principle #1 To understand the scope of the project:Scope of project provides the software team with a destination.

Principle #2 To involve the customer in the planning activity:The customer defines the software project priorities and establishes project constraints.To achieve this tasks software engineers must often negotiate timelines, order delivery, and other project related issues.

Principle #3 To recognize that planning is iterative:A software project plan is never engraved in stone.

Principle #4 To estimate based on what you know:The goal of estimation is to provide an indication of cost, efforts and task duration, based on the software project team’s current understanding of the work to be done.If information is unreliable, estimates will be equally reliable.

1. **Describe principles of testing for software engineering?**

Prinicple #1 All tests should be traceable to customer requirements: The main objective of software testing is to uncover errors therefore it follows that the most server defects are those that cause the program to fail to meet its requirements.

1. Principle #2 Test should be planned long before the testing begins: Testing planning can begin as soon as requirements model is complete.
2. Principle #3 The Pareto principle applies to software testing: In this context, the Pareto principle implies that 80% of all errors uncovered during testing will likely be traceable to 20% of all program components.
3. Principle #4 Testing should begin “in the small” and progress towards testing “in the large”: The first tests planned and executed generally focus in individual components.

Q 6. Describe principles of construction for software engineering?

The construction activity in software engineering encompasses a set of coding and testing tasks that lead to operational software that is ready for delivery to the customer.

1.Integration testing: Conducted as the system is constructed.

2.Validation testing: That assesses whether requirements have been met for the complete system.

3.Acceptance testing: That is conducted by the customer in an effort to exercise all required features and functions.

**7. Explain six functions of the requirement engineering process.?**

**8. Explain SRS and its advantage and disadvantages?**

The software requirements document called as software requirements specification or SRS is an official statement of what the system developers should implement.

It should include both the user requirements for a software system and a detailed specification of the system requirements.

It is a requirements specification for a software system, and show a complete description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software.

Advantages of SRS

Improved quality of the final product: By identifying and addressing requirements early on in the development process, using validation techniques can improve the overall quality of the final product.

Reduced development time and cost: By identifying and addressing requirements early on in the development process, using validation techniques can reduce the likelihood of costly rework later on.

Disadvantages of SRS

Increased time and cost: Using validation techniques can be time-consuming and costly, especially when involving multiple stakeholders.

Risk of conflicting requirements: Using validation techniques can lead to conflicting requirements, which can make it difficult to prioritize and implement the requirements.

1. **Write the characteristics and format of SRS?**

Complete: The SRS should include all the requirements for the software system, including both functional and non-functional requirements.

Consistent: The SRS should be consistent in its use of terminology and formatting, and should be free of contradictions.

Unambiguous: The SRS should be clear and specific, and should avoid using vague or imprecise language.

Traceable: The SRS should be traceable to other documents and artifacts, such as use cases and user stories, to ensure that all requirements are being met.

Verifiable: The SRS should be verifiable, which means that the requirements can be tested and validated to ensure that they are being met.

1. **Explain Requirement gathering and analysis**

Requirement Gathering:

Software requirements gathering is a subdomain of software requirement engineering

There are several techniques available for requirement gathering and most of them involve customer interaction with the development team.

The goal of the requirements-gathering activity is to collect all relevant information regarding the software to be developed from users’ requirements.

Requirement gathering combines all the aspects of problem-solving, elaboration, negotiation, and specifications.

Requirement Analysis:

Requirement analysis is the process of establishing the services that the customer requires from the system and the constraints under which is to be developed and operated.

Analysis is a process of understanding in detail what a system should accomplish.

Requirements analysis involves defining customer needs and objectives, environments and identified system characteristics to determine requirements for system functions.

IEEE defines requirements analysis as “the process of studying user needs to arrive at a definition of a system functions”. OR “the process of studying and refining system hardware or software requirements”.