Explain about safety and security of system systems

Safety:
Safety critical systems are systems where it is essential that system operation is always safe. That is the system should never damage people or the system's environment even if the system fails. Examples of safety critical system are control and monitoring systems in aircraft, process control process control systems in chemical and pharmaceutical plants and automobile control systems. Safety critical software are 2 types:

Primary safety-critical systems
Embedded software systems whose failure can cause hardware malfunction which results inhuman injury or environmental damage.

Secondary safety-critical systems
Systems whose failure indirectly results in injury. Eg - Medical Database holding details of drugs.

Ways to achieve Safety • Hazard avoidance • Hazard detection and removal • Damage limitation

Security:
Security is a system property that reflects the ability to protect itself from accidental or deliberate external attack.
Security is becoming increasingly important as systems are networked so that external access to the system through the Internet is possible. Security is an essential pre-requisite for availability, reliability and safety
Example: Viruses, unauthorised use of service/data modification.
Damage from insecurity:
Denial of service • Corruption of programs or data • Disclosure of confidential information

Write a short note on (i) Repository model (ii) Client Server Model (iii) Layered Model

A repository model

- It is a system that will allow interfacing sub-systems to share the same data.

Sub-system must exchange data so that they can work together effectively. This may be done in two ways:

i. All shared data is held in a central database that can be accessed by all subsystems. It is called repository model.
ii. Each sub-system maintains its own database.

Client-server model

It is a networking computing system design that illustrates a relationship between two or more computers, where the client computers request and receive services or resources from a powerful centralized server computer. It describes a specific way devices access the information you store in servers. It also allows multiple clients to open applications or retrieve files from an individual server, which helps maintain consistency across all devices. Many companies across various industries use servers to store and access information, offering more processing power and providing more extensive storage space.

Layerd Model:

The data link layer encapsulates data from the network layer, and the network layer encapsulates data from the transport layer

IP packets are transported in frames. To transport IP packets across all the various links, there are two theoretical options.

i. Change all the data link protocols so they understand IP addresses and the structure of IP packets.
ii. Use the data link protocols to transport IP packets as their data.

Explain user interface design process with diagram.

User interface is the front-end application view to which user interacts in order to use the software.
The software becomes more popular if its user interface is:

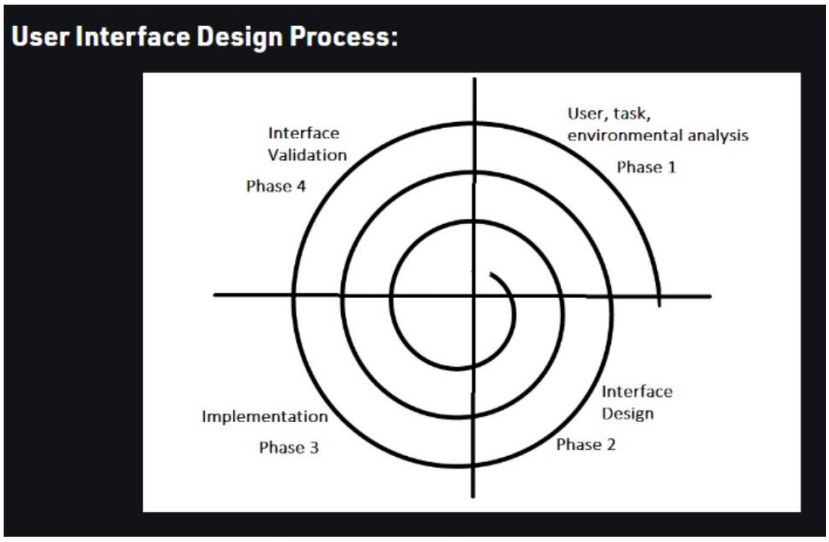
- Attractive
- Simple to use
- Responsive in short time
- Clear to understand
- Consistent on all interface screens

There are two types of User Interface:

i. Command Line Interface: Command Line Interface provides a command prompt, where the user types the command and feeds to the system. The user needs to remember the syntax of the command and its use.

ii. Graphical User Interface: Graphical User Interface provides the simple interactive interface to interact with the system. GUI can be a combination of both hardware and software. Using GUI, user interprets the software.

The analysis and design process of a user interface is iterative and can be represented by a spiral model.
The analysis and design process of a user interface is iterative and can be represented by a spiral model.



i. The goal of this phase is to define the set of interface objects and actions i.e. Control mechanisms that enable the user to perform desired tasks. Indicate how these control mechanisms affect the system.
ii. Specify the action sequence of tasks and subtasks, also called a user scenario.
iii. Indicate the state of the system when the user performs a particular task.

Explain the steps of Project scheduling.

What is a project schedule?
• A project schedule provides a general overview of your project, including the timeline, project tasks, dependencies, and assigned team members.
• Essentially, a project schedule should be able to tell you everything you need to know about your project at first glance.

7 steps to create a project schedule

• Define your project goals. Write down key milestones or deliverables that will make this project successful in the end.
• Identify all stakeholders. Make a list of every person that needs to interact with the project team, even if their role is a simple sign-off.
• Determine your final deadline. Decide when you need to be completely finished with the project. Be sure to give yourself enough time to account for conflicts or changes that might come up later during schedule management.
• List each step or task. Take those milestones and deliverables you defined in the first step and break them down into smaller tasks and subtasks to be sure all bases are covered.
• Assign a team member responsible for each task. Decide who will take on each task and subtask, and be transparent with deadlines. Remember that your colleagues likely have other projects going on at the same time. Be mindful of their workload so they don ’ t feel overloaded.

• Work backward to set due dates for each task. Figure out how long each task will take to complete (its start and end date), knowing that delays are inevitable. Sequencing is important to consider as well since certain tasks will need to be finished before another can start.
• Organize your project schedule in one tool, and share it with your team. You've successfully built your project plan and now it's important to organize it in a way that everyone involved can see and work from it. Finding a tool that can help you do both will be critical to your success.

Explain Requirement validation. What are its techniques?

Requirement validation

- Validation answers the question, "Are we building the right system?" Requirements validation is the process of checking that the defined requirements are for development, and defining the system that the customer really wants.
- Requirements validation helps us detect errors at an early stage of product development so that it does not result in excessive rework when detected later in the system development life cycle.
- In this blog post, we will look at what Requirements Validation is, its Process, and various Tools used for Requirements Validation.

Validation Techniques:

Checks:

- While checking the requirements, we proofread the requirements documents to ensure that no elicitation notes are missed out.
- During these checks, we also check the traceability level between all the requirements.
- For this, the creation of a traceability matrix is required.

- This matrix ensures that all the requirements are being properly considered and everything that is specified is justified.
- We also check the format of requirements during these checks.
- We also check the format of requirements during these checks.

Prototyping:

- This is a way of building a model or simulation of the system that is to be built by the developers.
- This is a very popular technique for requirements validation among stakeholders and users as it helps them to easily identify the problems, detect missing requirements and understand how technology can help them.
- We can just reach out to the users and stakeholders and get their feedback.

Test Design:

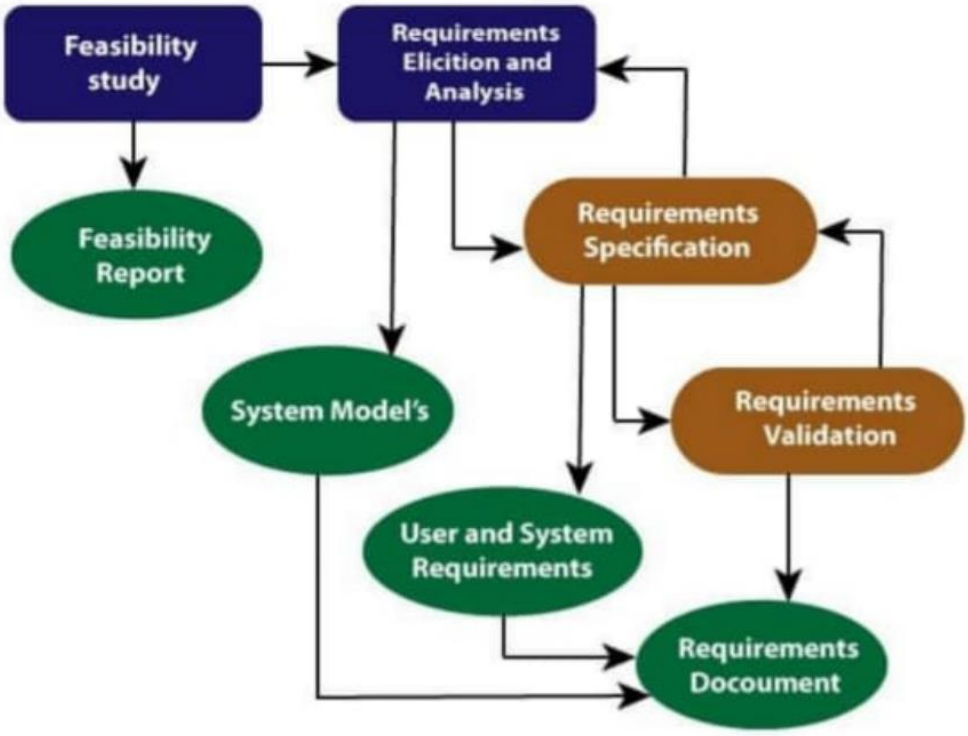
- During test designing, we follow a small procedure where the testing team build a few testing scenarios.
- Tests have to be derived from the requirements specification. The aim of this process is to figure out the errors in the specification or the details that are missed out leading to difficulties in the definition of the test scenarios.

Requirements Review:

- During requirement review, a group of knowledgeable people analyze the requirements in a structured and detailed manner and identify potential problems.
- After that, they gather up to discuss the issues and figure out a way to address the issues.
- A checklist is prepared that the reviewers fill up to provide a formal output of the review.
- After that, a final approval sign-off is done.

Explain Requirement engineering process

Requirement Engineering Process



Requirement Engineering Process

Requirements engineering is the process of identifying, eliciting, analyzing, specifying, validating, and managing the needs and expectations of stakeholders for a software system.

- The requirements engineering process is an iterative process that involves several steps, including:

Requirements Elicitation:

- This is the process of gathering information about the needs and expectations of stakeholders for the software system.
- This step involves interviews, surveys, focus groups, and other techniques to gather information from stakeholders.

Requirements Analysis:

- This step involves analyzing the information gathered in the requirements elicitation step to identify the high-level goals and objectives of the software system.
- It also involves identifying any constraints or limitations that may affect the development of the software system.

Requirements Specification:
- This step involves documenting the requirements identified in the analysis step in a clear, consistent, and unambiguous manner.
- This step also involves prioritizing and grouping the requirements into manageable chunks.

Requirements Validation:
- This step involves checking that the requirements are complete, consistent, and accurate.
- It also involves checking that the requirements are testable and that they meet the needs and expectations of stakeholders.

Requirements Management:
- This step involves managing the requirements throughout the software development life cycle, including tracking and controlling changes, and ensuring that the requirements are still valid and relevant.

The Requirements Engineering process is a critical step in the software development life cycle as it helps to ensure that the software system being developed meets the needs and expectations of stakeholders, and that it is developed on time, within budget, and to the required quality

List the types of Quality Assurance Standards. What is its importance?

- Quality assurance is the term used to describe the planned and systematic actions required to assure customers that contracted product characteristics and production processes are consistently delivered.
- Quality assurance involves the implementation of quality checks and standard operating procedures (SOP) that yield a consistent product and allows any issues to be identified and immediately corrected during production.
- Total Quality Management (TQM) is the term used to describe the management approach to long term success through customer satisfaction.
- Total Quality Management (TQM) is the term used to describe the management approach to long term success through customer satisfaction.
- Many organisations have developed their own standards for quality assurance whilst others have relied on public and private food safety and quality assurance standards.

The key objectives are:

- Ensuring maximum satisfaction of clients by meeting their quality requirements
- Safety of products and services during usage
- Complying with international regulations and local legislative rules
- Being environmentally responsible
- Confidentiality of stakeholders including customers, employees, partners, and investors
- Assuring a safer workplace for employees
- Optimum allocation of resources and minimisation of waste

Explain various control styles of software systems.

- Control models are models deployed in software engineering that are concerned with the control flow between the subsystems.
- They are distinct from the system decomposition model.

There are two types of control models: centralized and event-based control model.

1) Centralized Control Model
   - Centralized model is a formulation of centralized control in which one subsystem has overall responsibility for control and starts and stops other subsystems.
   - It is a control subsystem that takes responsibility for managing the execution of other subsystems.
   - Centralized models are classified into call-return and manager model.

2) Event-based Control Model
   - Event-based models are those in which each sub-system can respond to externally generated events from other subsystems or the system's environment.
   - It is a system driven by externally generated events where the timing of the events is out with the control of the subsystems which process the event.

   Event-based models are classified into broadcast and interrupt-driven models.