Date: YOUVA

Software Engineering
CIA - 1

Q1. What are the generic framework activities that are present in every software process?

Ans. Software processes generally consists of a set of frameworks activities that are essential for the development of software. The generic framework activities present in every software process are:-

Communication:
- Understand the problem and desired solution.
- Involve stakeholders, customers, developers and users.
- Establish ongoing feedback channels.

Planning:
- Define project scope, goals, milestones.
- Identify risks and plan how to deal with them.
- Allocate people, time, budget and resources.

Modeling:
- Create models for requirements, designs and analysis
- Visualize the system's structure and behavior.
- Facilitate communication among team members.

Constructing:
- Write, test and integrate software components.
- Translate design models into code.
- Validate code against requirements.

1. Deployment:

- Deliver the software to users.
- Install and configure in the target environment.
- Monitor system performance and usage.

Q2. Difference between prototype model and waterfall model with example.

Ans:- Waterfall Model :-

Working:

- Like climbing down a waterfall, one step at a time.
- Each phase is completed before moving to the next.

When to use:

- Best for projects with clear, unchanging requirements.
- Works well when everything is known for the start.

Customer involvement:

- Customers are mainly involved at the beginning and end.

Example:

- Building and basic website with fixed features, like a personal blog.

- Prototype Model :-

Working:

- Like building a rough model and improving it.
- It is flexible and involves creating an intial version and refining it based on feedback.

When to use:

- Good for projects where requirements are not clear or may change.

- Works well for creative projects with evolving ideas.

Customer involvement:

- Customers are actively involved throughout, providing feedback on prototypes.

Example:

- Developing a new mobile app with features that might change based on user feedback, like a social media app.

Q3. Discuss about risk model is beneficial or MNCs Projects. Explain its pros and cons for this model.

Ans. While a risk management model offers valuable benefits for MNCs projects, it is crucial to balance its use with projects needs and cultural considerations.

Pros:-

1. Better Problem Anticipation:
Helps identifying potential issues across ~~cultural~~ locations and cultures.

2. Informed Decision Making:
Guides decision making by assessing the likelihood and impact of risks.

3. Proactive Problem solving:
Encourages addressing issues early to avoid delays and complications.

4. Improved Communication:
Enhances communication among teams from diverse backgrounds.

5. Regulatory Compliance:
Ensures projects align with varied legal and regulatory requirements.

Cons :-

1. Resources Intensive:-
Requires time and effort, which may be challenging for tight schedules.

2. Excessive Documentation:
Could lead to too much paperwork, potentially diverting attention for actual work.

3. Uncertain Future Events:
Not all future events can be predicted accurately.

4. Cultural Challenges:
Implementation across different cultures may face communication hurdles

5. Resistance to Change:
Some team members may resist new practices, affecting implementation

Q4. Discuss customer myths about software development and their effect on the practitioner's performance as well as on overall outcome.

Ans. Common customer myths in software development :-

Myth: "Features can be added anytime."
Effect: Causes confusion and delays, affecting developer's focus.

Myth: "It should not take that long."
Effect: Unrealistic timelines lead to pressure, impacting morale and quality

Myth: "Do not need testing."
Effect: Ignoring testing leads to bugs and stress for testers.

Myth: "Adding more developers will speed up the project."
Effect: Overloading team hampers communication and productivity.

Myth: "Technology X is better than Technology Y and always."
Effect: Insisting on a specific tech without considering needs frustrates developers.

Myth: "We can cut costs by skipping Planning."
Effect: Skipping planning causes ~~theos~~ chaos, rework and missed deadlines.