

2 marks

Roll no: 2023178035

Name: Yogesh. M

① Software:

(*) Software is instructions (computer programs) that when executed provide desired features, function and performance and it is packed, with data structure that enables the program to adequately manipulate information.

② Types of changes in Legacy system:

Passes

(*) Legacy system evolves as time

requirements.

(i) Application to new technology
(ii) Enhanced to implement new

within evolving computing environment.

3) Suggest a few ways to build software to stop deterioration due to change:

(*) It is impossible to build a software to stop deterioration due to change, though there were several steps that a developer can follow to reduce deterioration.

- (i) Modular Design
- (ii) Abstraction
- (iii) Encapsulation
- (iv) Standardization
- (v) Continuous integration & deployment
- (vi) Refactoring

A) Elements of software process :

- (*) A process is a collection of
 - (i) Activities
 - (ii) Actions
 - (iii) Tasks

that are performed when some and work product is to be created.

B) Five generic process framework activities :

- (*) A generic process framework for software engineering encompasses five activities

- (i) communication
- (ii) Planning
- (iii) modelling
- (iv) construction
- (v) Deployment

6) Software engineering and WebApps :

Yes, software engineering principle is applicable to webapps. modifications includes UI/UX design , addressing client server architecture , utilizing web technology and framework , considering security , optimizing performance , ensuring browser compatibility , adopting CI/CD practices , managing data effectively and incorporating monitoring/ analytics .

7) Umbrella activities :

(*) In general umbrella activities are applied throughout a software project and help a software team manage and control progress , quality , change and risk . But their intensity and focus may vary across different framework activities .

8) Process flow :

(*) The process flow refers to the sequences of activities and task involved in the software development lifecycle . It describes how the framework activities and tasks that occur within each framework activity are organized w.r.t sequences , time .

9) Process pattern :

(*) A process pattern is a reusable approach or set of activities that address common challenges in software development, providing guidelines for organizing and executing tasks effectively.

10) Why does waterfall fails sometimes?

(*) Here are some reasons why waterfall model sometimes fail.

- (i) Rigidity
- (ii) Limited user involvement
- (iii) unclear Requirement
- (iv) long development time
- (v) Risk management
- (vi) Testing at the end
- (vii) Limited flexibility

11) Weakness of Evolutionary process model:

(*) The potential weakness of evolutionary process model involved uncertainty in timelines and budgets, scope, documentation challenges, testing complexity, resources allocation difficulties and managing stakeholders expectations.

- 12) XP Story :
- (*) An XP story is a simple description of a feature or functionality from the user's perspective, including acceptance criteria, used for planning and organising development tasks in Extreme Programming.
- 13) Two example of software projects to the incremental model :
- (*) Hospital management system.
(*) online learning platform.
- 14) Dual Role played by Software :
- (*) Product
(*) Vehicles
- 15) Software deterioration :
- (*) During its life, software will undergo change. As changes are made, it is likely that errors will be introduced, causing the failure rate curve to spike (actual curve). Before the curve can return to original steady-state failure rate, another change is requested, causing the curve to spike again slowly, the minimum failure rate level begins to

use the software is deteriorating due to change.

(b) Geographical separation and face to face communication in agile process model.

(*) Geographical separation in agile process doesn't imply avoidance, but it does present challenges, solutions include leveraging video conferences, regular virtual meeting, collaborative tools, documenting decisions, occasional in person meeting, cultural awareness and establishing clear communication system.

4 marks

i) How does a framework activity change as the nature of the project changes.

(*) A framework activity adapts to the changing nature of the project by adjusting its focus, intensity and execution approach.

Requirement analysis :

(*) In a project with well defined requirement this activity may focus more on validation and refinement. Conversely in projects with evolving requirements it may involve more exploration and iteration.

Design :

(*) For projects requiring extension innovations, the design activity may involve more prototyping and experimentation. In contrast projects with stable requirements may emphasize documentation.

Implementation :

(*) The implementation activity may vary based on project complexity for complex projects, it may involve incremental

development and continuous integration, while simpler projects may follow a straight forward coding process.

Testing and Deployment:

(*) Testing approaches may differ based on project constraints - for critical Projects, testing, with various technique may be crucial. Deployment activities may vary depending on the project scale and applicability.

Scale and applicability:

(*) Overall the framework activities adjust their emphasis, depth and execution strategies based on the unique characteristic constraints and gases of each Project.

Q) why formal methods were not widely used:

(*) Formal methods are capable of demonstrating software correctness, face limited adoption due to their complexity and high resources requirement. They are typically applied in safety-critical domains where correctness is paramount.

(*) However, for many projects, the practical challenges of implementation, such as tooling limitations and cultural resistance within development teams, outweigh the potential benefits. As a result, formal methods remain niche, approaches rather than mainstream practices in software development.

3) Provide 3 examples of software projects that would be amenable to the waterfall model.

(i) Simple website development:

(*) Building a basic website, like a portfolio site or an information website, follows a linear progression through requirements gathering, design, development, testing and deployment, making it suitable for the waterfall model.

(ii) Document management system:

(*) Developing a document management system for a small organization involves well-defined requirements that can be systematically addressed through sequential phases,

aligning with the waterfall model structured approach small - scale mobile app :

Creating a straightforward mobile app, such as a calculator or note-taking app, benefits from the waterfall model step-by-step progression allowing for systematic development and delivery within defined constraints.

A) moving outward along the spiral process flow:

You move outward along the spiral process flow, the software being developed or maintained typically undergo iterative refinement and enhancement. Each iteration represents a cycle of development, during which the software evolves incrementally.

With each cycle the software matures, features were added and modified and potential risks were addressed. As a result the software becomes increasingly robust, feature rich and aligned with stakeholder requirements.

5) combining process model :

(*) Yes, it is possible, to combine process model to tailor the development approach to the specific needs of a project. In this hybrid approach, initial phases such as requirements gathering and design may follow a sequential waterfall like process to establish a solid foundation.

(*) Once the design is in place, iterative and incremental development cycle, characteristic of agile methodologies, can be employed for implementation, testing and adaption based on user feedback. This hybrid model allows for a structured initial phase while incorporating the flexibility and responsiveness of agile methods throughout the development lifecycle.

6) what new practices are appended to XP to create IXP?

(*) In Industrial XP, specific practices are appended to traditional Extreme Programming to adopt it for industrial setting.

(i) Scaled agile framework (SAF):

(*) IXP incorporate SAF principle to manage dependencies, coordinate multiple teams and synchronize activities across the organisation, especially in larger projects.

(ii) Enterprise - level planning:

(*) IXP including practices for aligning projects goals with organizational objectives ensuring that development efforts contribute to business value and meet enterprise level requirements.

(iii) Risk management:

(*) IXP emphasizes robust risk management processes to identify, assess and mitigate risks associated with large-scale projects ensuring project success and stakeholder satisfaction.

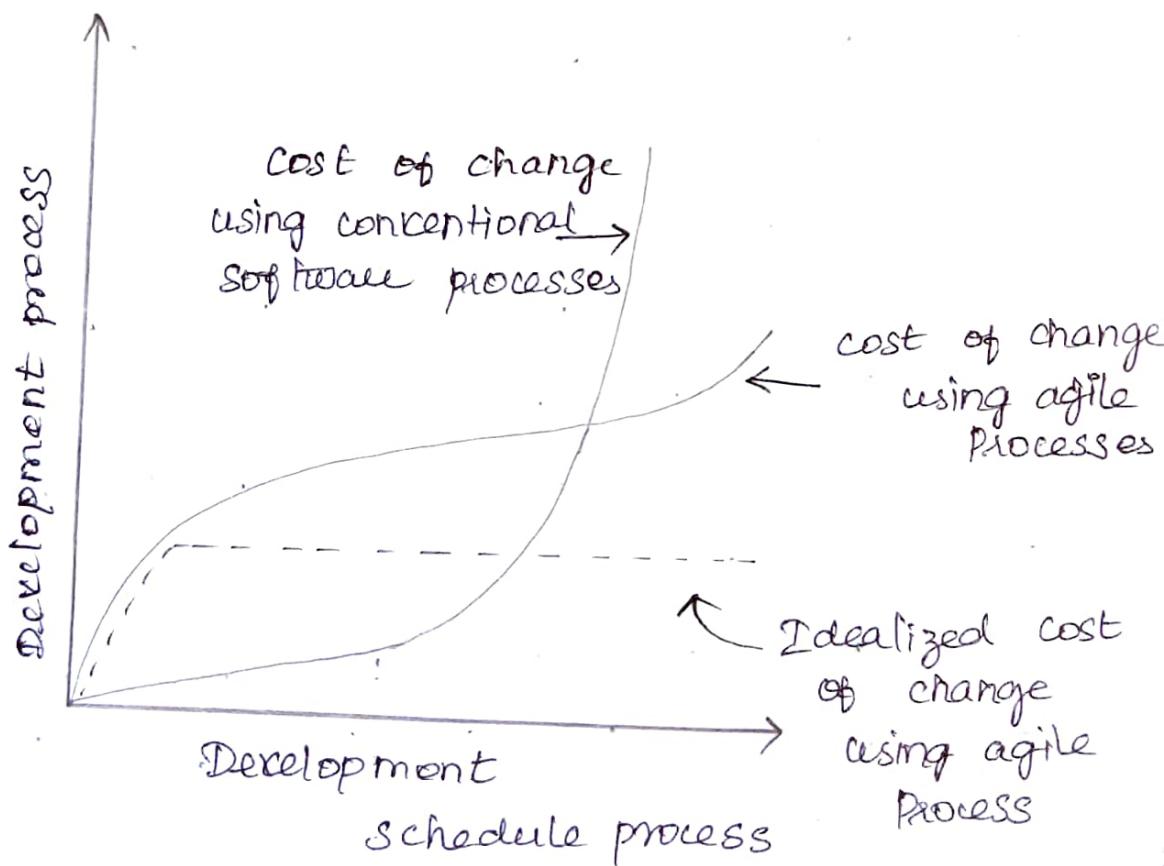
(iv) Continuous Integration and delivery(CI/CD):

(*) IXP emphasizes CI/CD practices to enable frequent and reliable delivery of software updates, enhancement and fixes, especially in complex and rapidly evolving development environment.

7) Agility and cost of change:

(a) Agility in software development refers to the ability to respond quickly and effectively to changes.

(b) The cost of change refers to the expenses or effort involved in making alterations to software after development. Agile practices aim to minimize the cost of change by enabling teams to adapt and deliver value incrementally throughout the development process, reducing the likelihood of costly late-stage modifications.



8) 12 principles of Agility :

- (i) customer satisfaction through early and continuous delivery of valuable software .
- (ii) welcome changing requirement even be in development agile process harness change for the customer's competitive advantage .
- (iii) Deliver working software frequently , with a preference for shorter timescales .
- (iv) Business people and developers must work together daily throughout the project .
- (v) Build projects around motivated individual , give them the environment and support they need and trust them to get the job done .
- (vi) The most efficient and effective method of conceiving information to and within a development team is face to face conversation .
- (vii) working software is the primary measure of process .
- (viii) Agile processes promote sustainable development the sponsors , developers and user should be able to maintain a constant

pace indefinitely.

- (ix) continuous attention to technical excellence and good design enhances agility.
- (x) simplicity , the art of maximizing the amount of work not done is essential.
- (xi) The best architecture , requirement, and design emerge from self-organizing teams.
- (xii) At regular intervals the team reflects on how to become more effective then tunes and adjust its behaviour accordingly .

These principle serve as the foundation for agile methodologies ; guiding teams is their approach to software development and project management .

8 marks

1) 5 Questions :

Question Designer should ask users :

1. What do you need this software to do for you?
2. what problem are you currently facing with your existing tools?
3. what features would make this software valuable to you?
A. Are there any specific preference or limitations we should consider?
4. How often will you use this software and in what situations?

Question user should ask Designers :

1. How will this software help me?
2. will it be easy to use?
3. can you show me similar project you worked on?
4. How will you incorporate my feedback?
5. will I have support when I need it?

Question user should ask themselves:

1. How do I want this software to accomplish?
2. How will benefit me?
3. Are there any risk involved in using IE?
4. How will it fit into my current workflow.
5. How will I know if its successful for me?

Questions Designer should Ask themselves:

1. How can we make this software meet user needs?
2. Is it easy to used and understand.
3. Are there any technical limitations we need to consider
4. How will we know if its working for users.
5. What steps can we take to improve it over time?

Q) Concurrent process model ?

(*) In the concurrent process model the states represent different stages or condition that a process can be in during its execution these states indicate what the process is currently doing and its progress towards completion.

(*) In the concurrent process and process can exist in several states including .

(i) Ready : The process is ready to execute but is waiting for the CPU to be allocated to it.

(ii) Running : The process is currently being executed by the CPU.

(iii) Blocked : the process is unable to proceed due to some external event such as waiting for input / output operations to complete or for a resources to become available

Terminal : The process has completed its execution.

These states come into play within the concurrent process model as processes interact with each other and with system resources. For example when a process is in the 'Blocked' state, the CPU can switch to executing another process that is in the ready state maximizing CPU utilization and overall system efficiency.

Additionally, the state transitions, such as moving from 'ready' to 'running' or from 'running' to 'blocked' are managed by the operating system to ensure efficient allocation of resources among concurrent processes.

3) Broad categories of computer software challenges for software engineers:

(i) complexity :

Dealing with the intricacies of large scale software systems,

Including managing interactions between component making interface logic, and maintaining comprehensible codebases.

(ii) Scalability :

Ensuring that software system can handle increasing workloads, data volumes and user interactions without sacrificing performance or reliability.

(iii) Security :

Protecting software system from various threats, including unauthorized access, data breaches, malware and other risks.

(iv) Reliability :

Building software that operates consistently and predictably under varying conditions, minimizing the occurrences of error, crashes and system failures.

(v) Maintainability :

Facilitating easy modification, enhancement and bug fixing of software.

system over-time ensuring that changes can be made without introducing unintended side effect.

(vi) compatibility :

Ensuring that software works correctly across different platforms, devices, operating system and environment allowing for seamless integration and interoperability.

(vii) Performance :

Optimizing software system to meet performance requirement, including response time, throughput and resource utilization to deliver a smooth and responsive user experience.

A) XP user story :

(*) user story : As a web user, I want have a browser to access and navigate various website so that I can browse the interest effectively

- 1) The browser should support basic navigation functionalities such as forward, backward, refresh and bookmarking.
- 2). It should provide a user-friendly interface for entering URL's and searching for content.
- 3) The browser must render web pages accurately maintaining proper layout formatting and multimedia content display.
- 4) It should support tabbed browsing allowing user to open multiple tabs
- 5) The browser should offer customized options such as themes, extension and setting for privacy and security preferences.
- 6) It should include features for managing browsing history, cookies and cached data.
- 7) The browser must be compatible with a range of operating system and

devices, ensuring accessibility for users across different platforms.

- 8) It should provide responsive and efficient performance with quick page loading time and minimal resource consumption.
- 9) The browser should offer integration with other services or applications, such as email clients, cloud storage and social media platforms.
- 10) It should prioritize user privacy and security, implementing measures to protect against malware, phishing attacks and data breaches.