

ASSIGNMENT-1

Q1 What is the main difference between programming and software engineering?

Ans) Programming vs Software Engineering

- It is the specific act of writing code. It is the discipline of applying engineering principles to the design, development, testing, and maintenance of software systems.

• A programmer focuses on the technical details of translating blueprints into functional codes. A software engineer works on the big picture, acting as an architect who oversees the entire software cycle.

• Programming is about writing code. Software engineering is when you take the pieces of code and consider how to adapt the code, how will task evolve etc.

Q2 What does the phrase "Hyrum's Law" state in software engineering?

Ans) Hyrum's Law states that with a sufficient number of users of an API, it doesn't matter what you promise in the contract: all observable behaviours of your system will be depended on by them.

③ Mention two reasons why long term projects need continuous upgrades?

- ans)
 - Hidden assumptions in new tasks
 - Lack of experience with legacy codes

Large upgrades often accumulate over years instead of being incremental.

④ What is meant by the phrase "shifting Left" in software development?

- ans) In software development, the phrase "Shifting Left" refers to the practice of performing key activities like testing, quality assurance and security checks earlier in the software development life cycle, that is, moving them "to the left" on the project timeline.

⑤ What is tradeoffs in engineering decisions?

- ans) In engineering, a trade-off is a compromise between competing factors. Improving one aspect (like speed) may worsen another (like cost or reliability). Engineers weigh these to find the best overall solution within constraints.

⑥ Mention two challenges faced by software engineers when working in teams.

ans) Many interpersonal issues in teams stem from a breakdown in humility, respect or trust.

- Communication gaps: Misunderstanding can arise due to unclear requirements, differing terminologies etc.

⑦ How does knowledge sharing benefit teams?

ans) Organizations understand their problem domain better than outsiders. This depends on building a culture of learning. Knowledge sharing helps to strengthen teams by:-

- i) Boosting efficiency
- ii) Increasing innovation.

iii) Enhancing collaboration . iv) Improving continuity.

⑧ Define the role of team lead.

ans) A team lead guides the team by setting goals, coordinating tasks, resolving conflicts, mentoring members and ensuring timely delivery of quality work.

⑨ Why team members should admit their mistake openly?

ans) Openly admitting mistakes fosters trust, accelerates problem solving and prevents small issues from escalating. It promotes a culture of learning and continuous improvement.

(11) Explain with example why a slow Bus factor is dangerous

for software projects?

ans) The "bus factor" measures how reliant a project is

on a single individual.

Slow bus factor means that only one or two developers understand critical parts of system. If they leave, get sick or are unavailable, the project is in serious trouble.

Eg.: Open-Source Software Library (Heartbleed bug, 2014)

which happened due to slow bus factors.

(12) Why is feedback culture important in software teams?

ans) Receiving feedback early in the development cycle can save time and prevent design flaws.

Feedback from diverse teammates adds depth and prevents tunnel vision.

Feedback from diverse teammates adds depth and prevents tunnel vision.