

1 Introduction - exam questions

You should know the answers to these questions:

How does Software Engineering differ from programming?

- Software Engineering involves a systematic, disciplined, and quantifiable approach to developing, operating, and maintaining software. Programming is just a subset, focusing on writing code.

Why is programming only a small part of the cost of a “real” software project ?

- Significant effort and cost come from requirements gathering, design, testing, maintenance, documentation, and quality assurance.

Give a definition for “traceability”.

- Traceability is the ability to relate components of a system back to their originating requirements to predict the impact of changes.

What is the difference between analysis and design?

- Analysis involves understanding and specifying “what” a system should do. Design focuses on “how” the system will achieve the requirements.

Explain verification and validation in simple terms.

- Verification ensures the system is built correctly (meets specifications). Validation ensures the system does what the user needs.

Why is the “waterfall” model unrealistic? Why is it still used?

- Unrealistic due to its rigid phases and difficulty accommodating changes. It’s used because of its simplicity and visibility for management.

Can you explain the difference between iterative development and incremental development?

- Iterative involves refining parts of the system through repetition. Incremental means delivering smaller, functional parts of the system progressively.

How do you decide to stop in the spiral model?

- You stop when all major risks have been resolved, and the project goals are met satisfactorily.

How do you identify risk? How do you assess a risk? Which risks require action?

- Risks are identified using checklists, projected by likelihood and impact, and prioritized. Action is taken on high-priority risks.

What is Failure Mode and Effects Analysis (FMEA)?

- FMEA is a method for identifying potential failures and analyzing their impact on the system.

List the 6 principles of extreme programming.

1. Fine-scale feedback
2. continuous process improvement
3. shared understanding
4. programmer welfare
5. coding
6. testing

What is a “sprint” in the SCRUM process?

- A sprint is a 2–4 week cycle during which a team develops a potentially shippable product increment.

Give the three principal roles in a scrum team. Explain their main responsibilities.

- **Product Owner:** Prioritizes the product backlog.
- **Scrum Master:** Facilitates the process and removes impediments.
- **Development Team:** Delivers the product increment.

Draw a UML class diagram modelling marriages in cultures with monogamy (1 wife marries 1 husband), polygamy (persons can be married with more than one other person), polyandry (1 woman can be married to more than one man) and polygyny (1 man can be married to more than one woman).

...

Draw a UML diagram that represents an object “o” which creates an account (balance initially zero), deposits 100\$ and then checks whether the balance is correct.

...

Can you answer the following questions?

What is your preferred definition of Software Engineering? Why?

- “State of the art of developing quality software on time and within budget” as it emphasizes both quality and real-world constraints.

Why do we choose “Correctness” & “Traceability” as evaluation criteria? Can you imagine some others?

- Correctness ensures the product meets user needs and specifications. Traceability predicts and manages the impact of changes.

Why is “Maintenance” a strange word for what is done during the activity?

- Maintenance involves enhancements, adapting to changes, and fixing defects, not just preserving functionality.

Why is risk analysis necessary during incremental development?

- Incremental development exposes new risks with each increment. Early identification ensures timely mitigation.

How can you validate that an analysis model captures users’ real needs?

- By involving users through prototyping, reviews, and feedback sessions.

When does analysis stop and design start?

- Analysis stops when the “what” is fully understood; design starts to define “how.”

When can implementation start?

- After completing sufficient design to provide clarity on system functionality and architecture.

Can you compare the Unified Process and the Spiral Model?

- Both are iterative, but the Spiral Model emphasizes risk management, while the Unified Process is more structured with defined phases.

Can you explain the values behind the Agile Manifesto?

- Focus on individuals and interactions, working software, customer collaboration, and responding to change.

Can you identify some synergies between the techniques used during extreme programming?

- Techniques like pair programming, test-driven development, and small releases reinforce quality, communication, and iterative learning.

Can you explain how the different steps in the scrum process create a positive feedback loop?

- Regular reviews, retrospectives, and planning allow teams to learn and improve continuously.

How does scrum reduce risk?

- By delivering increments early and often, ensuring transparency, and promoting adaptability to changes.

Is it possible to apply Agile Principles with the Unified Process?

- Yes, Agile's iterative nature and focus on adaptability align with Unified Process practices.

Did the UML succeed in becoming the Universal Modeling Language? Motivate your answer.

- Partially. UML is widely used but not universally adopted, as some industries prefer tailored modeling techniques.