Chapter 2

1. What is the primary focus of a software process?

- A) Testing and debugging
- B) Development and maintenance
- C) Sales and marketing
- D) Recruitment and hiring
- Answer: B

2. Which model is an example of a modern software process model mentioned in the text?

- A) Agile Model
- B) Rational Unified Process
- C) V-Model
- D) Spiral Model
- Answer: B

3. What does the 'specification' phase in software development involve?

- A) Implementing the system
- B) Defining what the system should do
- C) System testing
- D) Deployment of the system
- Answer: B

4. Which process model is known for its plan-driven approach?

- A) Incremental development
- B) Waterfall model
- C) Agile model
- D) Spiral model
- Answer: B

5. What is a key drawback of the waterfall model?

- A) Too flexible
- B) Difficult to accommodate change
- C) Too rapid
- D) Overemphasis on testing
- Answer: B

6. What is a major benefit of incremental development?

- A) High cost
- B) Longer time to deliver
- C) Reduced cost of changing customer requirements
- D) Increased complexity
- Answer: C

7. Which of the following best describes reuse-oriented software engineering?

- A) Developing from scratch
- B) Assembling systems from existing components
- C) Using least number of components
- D) Avoiding use of software libraries
- Answer: B

8. What is meant by 'system evolution'?

- A) Redesigning the system from scratch
- B) Maintenance and updating of the system
- C) Initial design phase
- D) Testing phase
- Answer: B

9. What is the primary role of software prototyping?

- A) Final system production
- B) Testing system security
- C) Demonstrating concepts and design options
- D) Long-term usage
- Answer: C

10. What type of development does Boehm's spiral model support?
    - A) Non-iterative development
    - B) Rapid prototyping
    - C) Risk-driven approach
    - D) Non-flexible development
    - Answer: C

11. In the context of the Rational Unified Process, what does the 'Inception' phase involve?
    - A) System testing
    - B) Establishing the business case for the system
    - C) Direct software development
    - D) Final deployment
    - Answer: B

12. Which is a characteristic of agile processes?
    - A) Comprehensive documentation
    - B) Emphasis on contract negotiation
    - C) Planning is incremental
    - D) Avoiding customer collaboration
    - Answer: C

13. What does 'change tolerance' refer to in software processes?
    - A) Avoiding any changes to the project plan

- B) Designing processes to easily accommodate changes
- C) Using the waterfall model exclusively
- D) None of the above
- Answer: B

14. Which phase involves converting system specification into an executable system?
    - A) Specification
    - B) Design and implementation
    - C) Validation
    - D) Evolution
    - Answer: B

15. What is the main purpose of 'system testing'?
    - A) To develop the system
    - B) To execute the system with test cases
    - C) To design the system's architecture
    - D) To deploy the system to the customer
    - Answer: B

16. Which model typically involves developing a prototype to show key features to customers?
    - A) Waterfall
    - B) Incremental
    - C) Spiral
    - D) All of the above
    - Answer: C

17. What does the 'validation' activity in software development involve?
    - A) Generating ideas for software
    - B) Testing the software

- C) Selling the software
- D) Planning the development stages
- Answer: B

18. How does incremental delivery benefit the software development process?
- A) By extending the development

time
- B) By reducing interaction with the customer
- C) By providing earlier system functionality to the customer
- D) By increasing the complexity of the system
- Answer: C

19. What is an outcome of the requirements engineering process?
- A) Reduced customer interaction
- B) Detailed system requirements specification
- C) Less documentation
- D) Faster system deployment
- Answer: B

20. In which software process activity are roles and responsibilities particularly emphasized?
- A) Design
- B) Specification
- C) Validation
- D) Process descriptions
- Answer: D

21. What is often a consequence of not incorporating change tolerance in software processes?
- A) Reduced costs

- B) Increased flexibility
- C) High costs of rework
- D) Faster development cycles
- Answer: C

22. Which process model emphasizes the reuse of software components?

- A) Waterfall
- B) Agile
- C) Reuse-oriented
- D) Spiral
- Answer: C

23. What does the Rational Unified Process emphasize for managing requirements?

- A) Ignoring customer feedback
- B) Documenting and tracking changes
- C) Reducing documentation
- D) Avoiding incremental development
- Answer: B

24. What is a benefit of software prototyping mentioned in the chapter?

- A) Creates complex systems
- B) Reduces the need for customer feedback
- C) Improves system usability and maintainability
- D) Increases the software development cost
- Answer: C

25. Which process model is characterized by a risk-driven approach?

- A) Incremental development
- B) Boehm's spiral model

- C) Rational Unified Process
- D) Waterfall model
- Answer: B

26. Which activity is NOT typically included in the software development process?

- A) Specification
- B) Validation
- C) Promotion
- D) Design
- Answer: C

27. What role does 'system evolution' play in software processes?

- A) It refers to the initial deployment of the system.
- B) It involves maintaining and modifying the system after its initial release.
- C) It is about reducing the software's functionalities.
- D) It focuses only on the system testing.
- Answer: B

28. Which statement is true about plan-driven processes?

- A) They avoid all forms of planning.
- B) They involve no documentation.
- C) They plan all activities in advance and measure progress against this plan.
- D) They are primarily used for small projects.
- Answer: C

29. What is the main advantage of reuse-oriented software engineering?

- A) It extends the software development time.
- B) It assembles the system from pre-existing components.

- C) It focuses on individual and isolated software development.
- D) It requires extensive testing and validation.
- Answer: B

30. In the context of software development, what is 'prototyping'?

- A) The final stage of software testing.
- B) The process of developing an initial version of a system to demonstrate concepts.
- C) A strategy to avoid user feedback.
- D) The development of detailed documentation.
- Answer: B

31. What does the 'waterfall model' traditionally emphasize?

- A) Flexibility in backtracking to earlier phases.
- B) Incremental development and delivery.
- C) Sequential completion of phases before moving to the next.
- D) Continuous integration of system components.
- Answer: C

32. How does 'incremental development' primarily handle changes in customer requirements?

- A) By ignoring changes once the development has started.
- B) Through rigidly structured development phases.
- C) By incorporating changes in subsequent increments.
- D) By reducing the software quality.
- Answer: C

33. Which of the following best describes the objective of 'change tolerance' in software processes?

- A) To prevent any changes to the original design.
- B) To facilitate easy incorporation of changes with minimal cost.

- C) To discourage customer feedback.
- D) To follow a strict plan without deviations.
- Answer: B

34. What is NOT a focus of the Rational Unified Process?
    - A) Iterative development
    - B) Risk assessment
    - C) Avoiding user involvement
    - D) Component-based architecture
    - Answer: C

35. What challenge does the waterfall model face in handling changes?
    - A) It is too flexible.
    - B) It handles changes seamlessly.
    - C) It finds accommodating changes difficult once a phase is complete.
    - D) It requires no planning.
    - Answer: C

36. What is a key feature of 'agile processes'?
    - A) Complete avoidance of user involvement.
    - B) Emphasis on comprehensive documentation.
    - C) Flexibility in adapting to changes.
    - D) Long-term planning is the core strategy.
    - Answer: C

37. Which method uses the development of prototypes to reduce the costs of rework?
    - A) Spiral model
    - B) Waterfall model
    - C) Incremental development

   - D) Both A and C
   - Answer: D

38. What is typically included in the 'validation' phase of software development?
   - A) Gathering user requirements.
   - B) Testing the system against customer requirements.
   - C) Designing the system architecture.
   - D) Promoting the software in the market.
   - Answer: B

39. Which approach would be best suited for a project with very stable requirements?
   - A) Agile development
   - B) Incremental development
   - C) Waterfall model
   - D) Prototyping
   - Answer: C

40. What does Boehm's spiral model integrate into the software development process?
   - A) Ignoring risks at all stages.
   - B) Sequential development without iteration.
   - C) Risk-driven approach with iterative refinement.
   - D) A single comprehensive development phase.
   - Answer: C

Chapter 3 on Agile Software Development

What is the primary role of the Scrum Master in a Scrum team?

A) To provide technical solutions to the team.

B) To facilitate Scrum processes and remove impediments.

C) To manage the project budget.

D) To write user stories for the product backlog.

Answer: B

What artifact provides a prioritized list of work or requirements for the Scrum team?

A) Sprint backlog

B) Product backlog

C) Scrum board

D) Burn-down chart

Answer: B

During which Scrum event do team members discuss what went well, what could be improved, and how to incorporate improvements into the next sprint?

A) Sprint Planning

B) Daily Scrum

C) Sprint Review

D) Sprint Retrospective

Answer: D

What is the purpose of the Daily Scrum?

A) To provide a status update to the product owner.

B) To discuss in detail the solutions to technical issues.

C) To inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary.

D) To assign tasks to team members.

Answer: C

How long is a Sprint typically in Scrum methodology?

A) One week

B) Two to four weeks

C) Six weeks

D) Three months

Answer: B

Which of the following best describes the role of the Product Owner in Scrum?

A) They are responsible for the Scrum framework adoption.

B) They maintain the Product Backlog and ensure that it is visible and clear to all.

C) They manage the development team.

D) They handle the technical implementation of the product.

Answer: B

What is the main focus of the Sprint Review?

A) To provide feedback on the product demo from the team.

B) To demonstrate the work done and gather feedback from stakeholders.

C) To assign tasks for the next sprint.

D) To evaluate the Scrum Master's performance.

Answer: B

What principle does Scrum emphasize by promoting self-organized teams?

A) Command and control

B) Team empowerment

C) Direct management intervention

D) Individual performance metrics

Answer: B

Which Scrum artifact helps track progress and forecast future work?

A) Product roadmap

B) Burn-down chart

C) User stories

D) Gantt chart

Answer: B

What does a Sprint Backlog consist of?

A) Items selected by the product owner only.

B) All items from the Product Backlog.

C) The set of Product Backlog items selected for the Sprint, plus a plan for delivering them.

D) Tasks that were not completed in the previous Sprint.

Answer: C

How does Scrum ensure transparency among the team members?

A) Through documentation

B) Through daily meetings and visible artifacts like the Scrum board

C) Through external audits

D) Through weekly reports to management

Answer: B

What is expected from the team at the end of a Sprint in Scrum?

A) A fully functioning increment of the product that meets the definition of done.

B) A complete product release.

C) A detailed report of individual activities.

D) An updated Gantt chart.

Answer: A

What concept does Scrum employ to manage scope and feature creep during a Sprint?

A) Timeboxing

B) Scope ballooning

C) Rolling wave planning

D) Feature stuffing

Answer: A

Which Scrum value emphasizes respect for people as individuals and professionals?

A) Commitment

B) Focus

C) Openness

D) Respect

Answer: D

How is work prioritized in the Scrum framework?

A) By the Scrum Master based on task complexity

B) By the development team based on skills

C) By the Product Owner based on business value

D) By external stakeholders based on market demands

Answer: C

1. What is a primary goal of Agile software development?

   - A) To follow a strict plan
   - B) To minimize change
   - C) To deliver working software quickly
   - D) To maximize documentation
   - Answer: C

2. Which of the following is NOT a value stated in the Agile Manifesto?

   - A) Comprehensive documentation over working software
   - B) Customer collaboration over contract negotiation
   - C) Responding to change over following a plan
   - D) Individuals and interactions over processes and tools
   - Answer: A

3. What does Extreme Programming (XP) emphasize?

   - A) Long-term planning
   - B) Frequent releases and iterative development
   - C) Complete up-front specifications
   - D) Minimal customer involvement
   - Answer: B

4. Which practice is commonly associated with XP?

   - A) Test-first development
   - B) Avoiding refactoring
   - C) Large initial design phases
   - D) Full documentation before coding
   - Answer: A

5. What is a key challenge in scaling Agile methods for large systems?

- A) Too rapid development
- B) Difficulty in maintaining frequent communication
- C) Simplified coding practices
- D) Reduced need for skilled developers
- Answer: B

6. Agile methods typically require what kind of team environment?

- A) Large, distributed teams without frequent communication
- B) Small, co-located teams with frequent communication
- C) Teams without any structured management
- D) Teams that follow a strict sequence of tasks
- Answer: B

7. Which statement best describes pair programming?

- A) Two programmers work separately on the same task for comparison.
- B) Two programmers work together at one workstation, continuously reviewing each other's work.
- C) Programmers work in pairs but on different components of the project.
- D) Programmers switch tasks every pair of hours to ensure versatility.
- Answer: B

8. What does the Scrum framework primarily focus on?

- A) Individual accountability and long-term planning
- B) Incremental, iterative development cycles called sprints
- C) Complete system testing before development
- D) Development without customer feedback
- Answer: B

9. In Agile methodologies, how are requirements typically handled?

- A) Fixed at the beginning of the project
- B) Changed only at designated milestones
- C) Continuously evolved and prioritized by the customer
- D) Defined by the development team
- Answer: C

10. What role does an 'on-site customer' play in XP?

- A) Provides occasional feedback
- B) Acts as a passive observer
- C) Is integrated into the development team for continuous feedback
- D) Reviews the project at completion stages only
- Answer: C

11. How does test-first development benefit software quality?

- A) By ensuring that documentation is prioritized
- B) By writing tests after software features are developed
- C) By developing tests before the software feature to clarify requirements
- D) By eliminating the need for testing
- Answer: C

12. What is the principle challenge of Agile methods in large organizations?

- A) Too rapid development cycles
- B) Integration with existing bureaucratic processes
- C) Lack of need for skilled developers
- D) Overemphasis on documentation
- Answer: B

13. Which of the following is NOT a principle of Agile software development?

- A) Emphasize process over people
- B) Embrace change
- C) Incremental delivery
- D) Customer involvement
- Answer: A

14. What does the Scrum master do?

- A) Writes the software code
- B) Facilitates the team processes and shields the team from external distractions
- C) Acts as the chief architect
- D) Is responsible for performing all the testing
- Answer: B

15. Which is a benefit of Agile project management?

- A) Strict adherence to predetermined plans
- B) Flexibility and rapid response to change
- C) Complete automation of software development
- D) Elimination of all documentation
- Answer: B

16. Why is refactoring important in Agile development?

- A) It discourages

change in the development process

- B) It simplifies the code, making it easier to adapt and modify
- C) It prevents any new features from being added
- D) It ensures that the original design is never altered
- Answer: B

17. What does the Agile principle 'people over process' emphasize?

- A) The irrelevance of having skilled developers
- B) The importance of following a strict plan
- C) The skills and ideas of individuals over rigid processes
- D) The elimination of all project management practices
- Answer: C

18. How do Agile methods enhance software maintenance?

- A) By avoiding any maintenance tasks
- B) By reducing the documentation needed for maintenance
- C) By making the system easier to adapt and extend
- D) By fully documenting every detail before development
- Answer: C

19. What challenge do Agile teams face when involving customers in the development process?

- A) Over-involvement of customers can slow down the process
- B) Customers are often reluctant to provide timely feedback
- C) It can be difficult to maintain customer interest over time
- D) Customers often want to use a plan-driven approach
- Answer: C

20. Which of the following best describes incremental development in Agile?

- A) Development occurs in small, manageable increments that are frequently delivered
- B) The entire software is developed in one increment
- C) Development occurs in increments but only once a year
- D) Increments are planned but not executed
- Answer: A

21. Which Agile practice involves writing tests before the actual software code?

- A) Test-driven development
- B) Continuous integration
- C) Scrum planning
- D) Sprint reviews
- Answer: A

22. What is a key benefit of using Scrum in project management?

- A) It eliminates the need for regular meetings.
- B) It allows for a rigid, unchanging plan.
- C) It facilitates flexible responses to changes.
- D) It ensures that no testing is required.
- Answer: C

23. In Extreme Programming, what does 'pair programming' facilitate?

- A) Decreased software quality
- B) Increased individual ownership of code
- C) Knowledge sharing and improved code quality
- D) Slower development speeds
- Answer: C

24. What challenge is associated with Agile methods in terms of team composition?

- A) Only senior developers can participate.
- B) Teams need a mix of highly skilled developers.
- C) Teams should not communicate frequently.
- D) Agile methods work with any team, regardless of skill level.
- Answer: B

25. Which statement about Agile documentation is true?

- A) Agile projects require more documentation than plan-driven projects.
- B) Agile projects do not produce any documentation.
- C) Agile projects focus on minimal documentation necessary for the task.
- D) Documentation is entirely replaced by automated tests.
- Answer: C

26. How does Scrum handle project tasks?

- A) Through a series of phases known as sprints, each typically lasting 2-4 weeks
- B) By completing all tasks in a single long phase
- C) By delegating all tasks to the Scrum Master
- D) Scrum does not include task management.
- Answer: A

27. What role does the customer play in Extreme Programming (XP)?

- A) The customer is only involved at the project's start.
- B) The customer is part of the team, continuously interacting and providing input.
- C) The customer's role is limited to final approval.
- D) Customers are typically discouraged from interacting with the development team.
- Answer: B

28. What is a primary focus of Agile project management?

- A) Following a detailed, long-term plan
- B) Adapting to changes and delivering functional software in short cycles
- C) Avoiding customer feedback to streamline development
- D) Maximizing the number of features per release
- Answer: B

29. Which of these is a practice unique to Extreme Programming?

- A) Daily stand-ups

- B) Pair programming
- C) Use of a product backlog
- D) Phase gate reviews
- Answer: B

30. How does refactoring contribute to Agile development?
    - A) By permanently fixing the initial design early in development
    - B) By frequently improving the design of the code throughout the project
    - C) By reducing the frequency of code reviews
    - D) By increasing the complexity of the code over time
    - Answer: B

31. What does 'sustainable pace' mean in the context of XP?
    - A) Working long hours to meet deadlines
    - B) Maintaining a work pace that avoids burnout and can be sustained indefinitely
    - C) Rapidly accelerating the pace of development at the end of each cycle
    - D) Slowing down the development process to ensure quality
    - Answer: B

32. Why is continuous integration important in Agile methods?
    - A) It prevents any form of integration until the end.
    - B) It allows the team to detect issues early by integrating and testing work frequently.
    - C) It is only used at the beginning of the project.
    - D) It discourages frequent testing.
    - Answer: B

33. What does the Scrum Master do during a Sprint?
    - A) Writes the majority of the code
    - B) Facilitates the process and removes obstacles

- C) Acts solely as a project accountant

- D) Is typically the main decision-maker on technical issues

- Answer: B

34. How are requirements managed in Agile methods like XP?

- A) They are fully specified before development starts and remain unchanged.

- B) They are continuously developed and prioritized through collaboration with the customer.

- C) They are considered only at the end of the development process.

-

D) They are irrelevant; Agile does not consider requirements.

- Answer: B

35. What is an Agile approach to handling project changes?

- A) Rejecting all changes once the project starts

- B) Accepting changes at any time during the project to better meet customer needs

- C) Only accepting changes at the end of each month

- D) Changes are never allowed in Agile projects.

- Answer: B

36. Which is a benefit of test-first development in Agile?

- A) It eliminates the need for any testing.

- B) It clarifies requirements before development, reducing rework.

- C) It prolongs the testing phase to ensure quality.

- D) It is used only at the project's completion.

- Answer: B

37. What is the impact of pair programming on code quality?

- A) It decreases code quality due to frequent disagreements.
- B) It improves code quality through continuous code review and collaboration.
- C) It has no impact on code quality.
- D) It reduces code quality by slowing down development.
- Answer: B

38. In Scrum, what is the purpose of the product backlog?

- A) It is a detailed plan for the entire project lifecycle.
- B) It lists all completed tasks.
- C) It contains all tasks and features awaiting development.
- D) It is only reviewed at the end of the project.
- Answer: C

39. How does Agile deal with project documentation?

- A) Produces as much documentation as possible
- B) Produces minimal, necessary documentation to support development
- C) Completely eliminates documentation
- D) Documentation is outsourced
- Answer: B

40. What challenge do distributed Agile teams face?

- A) No challenges, as Agile is designed for distributed teams
- B) Communicating effectively across different locations
- C) Having too many team members
- D) Implementing changes too quickly
- Answer: B

Based on Chapter 4 → System requirements

1. What is the primary purpose of defining functional requirements?

- A) To determine the budget of a project.
- B) To specify what the system should do.
- C) To establish the project timeline.
- D) To design the user interface.
- Answer: B

2. Which method is commonly used for gathering functional requirements?

- A) Performance testing
- B) Requirement elicitation
- C) Code refactoring
- D) Software prototyping
- Answer: B

3. What role do scenarios play in functional requirements?

- A) They provide specific instances of how a user might interact with the system.
- B) They outline the system's installation process.
- C) They specify the hardware requirements.
- D) They list potential defects in the system.
- Answer: A

4. In what document are functional requirements typically detailed?

- A) The user manual
- B) The system specification document
- C) The test plan
- D) The marketing plan
- Answer: B

5. What is the significance of 'use cases' in functional requirements?

- A) They determine the system’s non-functional requirements.
- B) They describe the sequence of events that a system performs.
- C) They provide data flow diagrams.
- D) They outline the financial budget of the project.
- Answer: B

6. What is a key characteristic of a good functional requirement?
   - A) Broad and ambiguous
   - B) Clear and concise
   - C) Complicated and detailed
   - D) Non-specific but elaborate
   - Answer: B

7. How are functional requirements validated?
   - A) Through user acceptance testing.
   - B) By reviewing the project budget.
   - C) By assessing the project timeline.
   - D) Through performance benchmarks.
   - Answer: A

8. What type of diagram is often used to model functional requirements?
   - A) Gantt chart
   - B) Use case diagram
   - C) Pie chart
   - D) Network diagram
   - Answer: B

9. Which stakeholder typically provides input for functional requirements?
   - A) Project sponsors

- B) End-users
- C) Competitors
- D) Regulatory bodies
- Answer: B

10. What is an outcome of poorly defined functional requirements?
    - A) Increased development speed
    - B) Lower overall project costs
    - C) Increased risk of project failure
    - D) Enhanced system security
    - Answer: C

11. Which activity is a part of functional requirement analysis?
    - A) Cost-benefit analysis
    - B) Structural analysis
    - C) Requirement feasibility
    - D) Security auditing
    - Answer: C

12. What does a functional requirement specification include?
    - A) Descriptions of system behavior under specific conditions.
    - B) The color scheme of the software interface.
    - C) The resumes of developers.
    - D) The physical dimensions of the hardware.
    - Answer: A

13. How do changes in functional requirements impact software projects?
    - A) They decrease documentation needs.
    - B) They may lead to scope creep.

- C) They reduce development time.
- D) They decrease testing requirements.
- Answer: B

14. What is an example of a functional requirement?
    - A) The system shall generate invoices.
    - B) The software must load in under two seconds.
    - C) The application shall be coded in Java.
    - D) The system shall be available 99.9% of the time.
    - Answer: A

15. Why is prioritization important in managing functional requirements?
    - A) It ensures the most critical features are developed first.
    - B) It helps in selecting the programming language.
    - C) It aids in marketing the product.
    - D) It determines the software's color scheme.
    - Answer: A

16. What does traceability refer to in the context of functional requirements?
    - A) The ability to interconnect different software systems.
    - B) The ability to trace the origin and impact of each requirement throughout the project.
    - C) The software's ability to trace user actions.
    - D) Tracking changes in software performance over time.
    - Answer: B

17. How are functional requirements often prioritized?
    - A

) Based on their complexity.

- B) Based on their impact on the business.
- C) Based on the color preferences of stakeholders.
- D) Based on the availability of technology.
- Answer: B

18. Which tool is NOT typically used to manage functional requirements?
    - A) Requirements management software
    - B) Database management systems
    - C) Text editors
    - D) Graphics design software
    - Answer: D

19. What aspect of software development is directly influenced by functional requirements?
    - A) Budget allocation
    - B) Feature development
    - C) Logo design
    - D) Office layout
    - Answer: B

20. Which phase in software development heavily involves functional requirements?
    - A) Initial planning
    - B) Mid-project review
    - C) Post-deployment
    - D) Pre-sale marketing
    - Answer: A

21. What is a common challenge when managing functional requirements?
    - A) Communicating changes to all stakeholders.
    - B) Choosing an appropriate font for documentation.

- C) Deciding on the software’s name.
- D) Selecting office furniture.
- Answer: A

22. Which factor is crucial for the successful implementation of functional requirements?
   - A) The aesthetic appeal of the user interface.
   - B) The physical location of the development team.
   - C) Clear and unambiguous documentation.
   - D) The personal preferences of the project manager.
   - Answer: C

23. What does the completeness of a functional requirement ensure?
   - A) It covers all possible user interactions with the system.
   - B) It specifies minimum system downtime.
   - C) It includes all legal disclaimers.
   - D) It outlines all potential marketing strategies.
   - Answer: A

24. How does automation impact the management of functional requirements?
   - A) It eliminates the need for requirements.
   - B) It facilitates tracking and consistency checks.
   - C) It decides the software’s pricing strategy.
   - D) It selects the project’s stakeholders.
   - Answer: B

25. What is the impact of ambiguous functional requirements?
   - A) They streamline the development process.
   - B) They lead to clear and concise documentation.
   - C) They may result in software that does not meet user expectations.

- D) They reduce the time needed for user acceptance testing.
- Answer: C

26. In what way do functional requirements interact with non-functional requirements?

- A) They dictate the non-functional requirements.
- B) They are completely independent of non-functional requirements.
- C) They are typically less important than non-functional requirements.
- D) They must be consistent with non-functional requirements to ensure system coherence.
- Answer: D

27. Which activity helps in refining functional requirements?

- A) User feedback and prototype testing.
- B) Selecting project management tools.
- C) Deciding the project's geographical location.
- D) Choosing corporate sponsors.
- Answer: A

28. What is a consequence of not involving end-users in the functional requirement gathering process?

- A) The project will be completed ahead of schedule.
- B) The software may not fully address user needs.
- C) The software will have enhanced security features.
- D) The development costs will be minimized.
- Answer: B

29. What strategy is effective in managing changes to functional requirements?

- A) Avoiding any changes to the requirements once they are defined.
- B) Implementing a flexible change management process.
- C) Ignoring stakeholder feedback.

- D) Limiting the documentation of requirements.
- Answer: B

30. Which of the following best describes the validation of functional requirements?

- A) Checking if the requirements meet aesthetic standards.
- B) Ensuring the requirements accurately reflect the needed system functionality.
- C) Confirming that the software uses the latest technology.
- D) Verifying that the requirements are popular among users.
- Answer: B

Chapter 5 System Modeling

1. What is the purpose of system modeling?

- A) To code software applications directly.
- B) To develop abstract models of a system from different perspectives.
- C) To document the financial aspects of systems.
- D) To test the system physically.
- Answer: B

2. What does UML stand for?

- A) Unified Modeling Language.
- B) Universal Modeling Language.
- C) User Mode Logic.
- D) Unified Method Language.
- Answer: A

3. Which diagram is NOT a type of UML diagram?

- A) Activity diagram.
- B) Sequence diagram.

- C) Flowchart diagram.
- D) Class diagram.
- Answer: C

4. What is a context model used for in system modeling?

- A) To show the internal structure of the system.
- B) To illustrate the operational context of the system and its environment.
- C) To demonstrate the dynamic behavior of the system.
- D) To code software directly from the model.
- Answer: B

5. Which perspective is concerned with the interactions between a system and its environment?

- A) External perspective.
- B) Interaction perspective.
- C) Structural perspective.
- D) Behavioral perspective.
- Answer: B

6. What is the main focus of structural models?

- A) Interactions between systems.
- B) The organization of a system or the structure of the data it processes.
- C) The dynamic behavior of the system.
- D) The graphical user interface of the system.
- Answer: B

7. Which UML diagram is used to show the interactions that occur between system components?

- A) Class diagram.
- B) State diagram.

- C) Sequence diagram.

- D) Use case diagram.

- Answer: C

8. What does a class diagram primarily depict?

- A) The sequence of operations in a system.

- B) The interaction between users and the system.

- C) The object classes in the system and relationships between them.

- D) The response of the system to various events.

- Answer: C

9. What is model-driven engineering focused on?

- A) Generating code manually from detailed documentation.

- B) Developing software without any models.

- C) Generating a complete or partial system implementation from the system model.

- D) Ignoring all modeling conventions.

- Answer: C

10. What type of models are used during requirements engineering to clarify existing system functions?

- A) Future system models.

- B) Existing system models.

- C) Behavioral models.

- D) Static system models.

- Answer: B

11. Which UML diagram type is specifically used to model the dynamic behavior of a system?

- A) State diagrams.

- B) Class diagrams.

- C) Activity diagrams.
- D) Component diagrams.
- Answer: A

12. In system modeling, what does the external perspective focus on?
    - A) Internal system functions only.
    - B) The structure of the database.
    - C) The environment or context of the system.
    - D) The detailed programming within the system.
    - Answer: C

13. How are interaction models useful?
    - A) They show static relationships only.
    - B) They help understand system performance and user requirements.
    - C) They are not used in modern system modeling.
    - D) They focus solely on the system's downtime.
    - Answer: B

14. Which is an example of using a model for discussion purposes?
    - A) Using a fully detailed and correct model for system implementation.
    - B) Using an incomplete model to facilitate discussion about a proposed system.
    - C) Using a model only after it has been fully implemented and tested.
    - D) Never using models in discussions.
    - Answer: B

15. What does a 'use case' in UML represent?
    - A) A diagram showing the physical layout of a system.
    - B) A dynamic model of system behavior.
    - C) A discrete task involving external interaction with a system.

- D) The programming behind a system.
- Answer: C

16. Why are sequence diagrams important?

- A) They depict the static structure of a system.
- B) They show the sequence of interactions within a system for a specific use case.
- C) They are used to model the user interface only.
- D) They represent the financial model of a system.
- Answer: B

17. What role do structural models play in system design?

- A) They are used to understand the system's interactions only.
- B) They help discuss and design the system architecture.
- C) They are only used after the system is built.
- D) They focus on external events affecting the system.
- Answer: B

18. Which model type would you use to describe the part-of relationship in semantic data models?

- A) Behavioral models.
- B) Aggregation models.
- C) Interaction models.
- D) External models.
- Answer: B

19. What does an activity diagram in UML show?

- A) The static structure of data.

- B) The activities involved in a process or data processing.

- C) The detailed code within each method.

- D) The financial transactions within a system.

- Answer: B

20. Which type of modeling would be used to show how a system responds to internal and external events?

- A) Data-driven modeling.

- B) Event-driven modeling.

- C) Process-driven modeling.

- D) Function-driven modeling.

- Answer: B

21. What is the primary benefit of model-driven engineering (MDE)?

- A) It relies on manual coding for system implementation.

- B) It allows for automatic code generation from models, raising the level of abstraction.

- C) It emphasizes extensive manual documentation.

- D) It eliminates the need for testing the software.

- Answer: B

22. What is a generalization in object-oriented modeling?

- A) It defines the specific attributes of a single class.

- B) It refers to a detailed description of every function.

- C) It involves creating a hierarchy where more general classes share attributes with more specific classes.

- D) It involves breaking down complex systems into simpler ones without shared attributes.

- Answer: C

23. Which UML diagram helps in understanding the sequence of activities within a system?

- A) Class diagram

- B) State diagram

- C) Activity diagram

- D) Use case diagram

- Answer: C

24. In the context of UML, what does aggregation represent?

- A) A behavioral relationship between classes.

- B) A computational method within a class.

- C) A part-of relationship between classes.

- D) An external interaction with the system.

- Answer: C

25. What advantage does the use of state machine models provide in system modeling?

- A) They primarily show static relationships.

- B) They illustrate the system's behavior in response to external and internal events.

- C) They depict the physical architecture of the system.

- D) They describe how to code the system in a specific programming language.

- Answer: B

26. What is the role of UML in system modeling?

- A) To provide a standardized graphical notation for representing various aspects of a system.

- B) To replace traditional programming languages.

- C) To focus solely on the financial aspects of systems.

- D) To document user requirements without diagrams.

- Answer: A

27. How are use cases integrated into system modeling?

- A) By programming the backend logic of systems.

- B) By visually representing discrete tasks that involve external interaction with the system.
- C) By depicting the dynamic relationships within the programming code.
- D) By modeling the hardware requirements of the system.
- Answer: B

28. Why are behavioral models important in system modeling?
   - A) They document the static attributes of the system.
   - B) They are used only during the initial stages of development.
   - C) They model the dynamic behavior of the system as it responds to different stimuli.
   - D) They eliminate the need for any other form of modeling.
   - Answer: C

29. What is an example of applying model-driven engineering in software development?
   - A) Completely avoiding the use of models.
   - B) Using detailed textual descriptions instead of graphical models.
   - C) Generating executable code directly from high-level models.
   - D) Focusing exclusively on manual coding practices.
   - Answer: C

30. What does a sequence diagram primarily show?
   - A) The physical components of a system.
   - B) The sequence of interactions within a system for a particular use case.
   - C) The static data structure of the system.
   - D) The financial transactions processed by the system.
   - Answer: B

Based on the content of Chapter 6 on "Architectural Design,"

1. What is the primary purpose of architectural design in software engineering?

   - A) To determine the project budget

   - B) To identify sub-systems and their interactions within a system

   - C) To write detailed software code

   - D) To handle marketing for software products

   - Answer: B

2. What does 'architecture in the large' refer to?

   - A) The architecture of small, individual software programs

   - B) The architecture concerning the organization of complex systems across multiple platforms

   - C) The physical layout of network components

   - D) The use of large data structures and algorithms

   - Answer: B

3. What is a key benefit of having an explicit software architecture?

   - A) It simplifies the code-writing process

   - B) It facilitates communication among stakeholders and supports system analysis

   - C) It reduces the need for software testing

   - D) It eliminates the need for project management

   - Answer: B

4. What are 'architectural patterns'?

   - A) Preset code libraries used in programming

   - B) Templates for documenting software projects

   - C) Stylized descriptions of good design practices tested in different environments

   - D) A type of software testing method

   - Answer: C

5. Which architectural pattern separates presentation and interaction from the system data?

- A) Layered architecture
- B) Client-server
- C) Model-View-Controller (MVC)
- D) Repository
- Answer: C

6. What is the primary focus of the 'Layered architecture' pattern?

- A) To facilitate user interface design
- B) To organize the system into a set of layers with each providing services to the next
- C) To manage database transactions only
- D) To implement direct communication between software components
- Answer: B

7. In which scenario is the Repository architectural pattern most appropriate?

- A) When systems require frequent, direct communication between components
- B) Where large volumes of data need to be accessed by numerous sub-systems
- C) When the system has no need for data persistence
- D) For systems that are purely computational without data storage
- Answer: B

8. How does the Client-server architecture typically distribute functionality?

- A) By using a single centralized server for all operations
- B) Through a decentralized arrangement where all components are equal
- C) By dividing functionality into services provided by separate servers accessed by clients
- D) All components operate independently without networking
- Answer: C

9. What is an advantage of using a pipe and filter architecture?

- A) It does not require data to be in any specific format

- B) It supports easy modification and reusability of processing components

- C) It is best suited for real-time interaction and user interfaces

- D) It simplifies security management by minimizing data transformations

- Answer: B

10. What does the term '4 + 1 view model of software architecture' refer to?

- A) A single model that combines four different architectures into one

- B) Four primary views (logical, process, development, physical) integrated through scenarios

- C) Four separate models used at different stages of software development

- D) A new software development methodology

- Answer: B

11. Which architectural view focuses on how a system is decomposed for development?

- A) Logical view

- B) Process view

- C) Development view

- D) Physical view

- Answer: C

12. What is a critical factor to consider when choosing an architectural style?

- A) The color scheme of the user interface

- B) The programming language to be used

- C) The non-functional requirements of the system

- D) The personal preferences of the development team

- Answer: C

13. Which statement best describes the impact of architectural decisions on system characteristics?

- A) They only affect the system's operational speed.
- B) They influence non-functional characteristics such as security and performance.
- C) They are irrelevant to the system's scalability and maintenance.
- D) They solely determine the aesthetic aspects of the system interface.
- Answer: B

14. What is the role of 'architectural views'?

- A) To provide different perspectives of the

system for better understanding and communication

- B) To focus only on the technical implementation details
- C) To outline the financial aspects of the system
- D) To describe the legal framework surrounding the system development
- Answer: A

15. What does the MVC pattern primarily manage?

- A) Data storage and database management
- B) Separation of presentation, control, and data management concerns
- C) Network configurations and security
- D) User authentication and authorization
- Answer: B

16. How do layered architectures improve system maintainability?

- A) By mixing all system functions into a single layer
- B) By allowing changes in one layer without affecting others
- C) By eliminating the need for documentation
- D) By focusing only on the user interface
- Answer: B

17. What is a common use of the Repository architectural pattern?

- A) To enhance graphical performance
- B) To manage shared data accessed by different subsystems
- C) To reduce the storage requirements of the system
- D) To simplify direct data manipulation by end-users
- Answer: B

18. Which pattern is specifically structured to handle data transactions and storage efficiently?

- A) MVC
- B) Pipe and filter
- C) Client-server
- D) Repository
- Answer: D

19. Why might an organization choose a client-server architecture?

- A) To avoid data replication
- B) To centralize all processing in one location
- C) For efficient resource sharing and service distribution
- D) To ensure that no networking is required
- Answer: C

20. Which architectural pattern is characterized by its use of layers to provide system services?

- A) MVC
- B) Layered
- C) Client-server
- D) Pipe and filter
- Answer: B

21. What is a disadvantage of the client-server model?

- A) It simplifies the management of services
- B) Servers can be a single point of failure
- C) It eliminates the need for a network
- D) It ensures that data is never replicated
- Answer: B

22. What benefit does the MVC architectural pattern offer?

- A) It combines data handling and user interface in one component
- B) It allows changes in the user interface without affecting data handling
- C) It prevents users from interacting with the system directly
- D) It minimizes the security of the system
- Answer: B

23. In a layered architecture, how is data typically handled between layers?

- A) Data is not shared between layers to enhance security
- B) Each layer provides services to the layer above it
- C) All layers operate independently without any interaction
- D) Data handling is confined to the topmost layer only
- Answer: B

24. Why is the pipe and filter architecture advantageous for data processing applications?

- A) It requires minimal data formatting
- B) Each filter component can be independently developed and reused
- C) It is ideal for real-time interactive applications
- D) It focuses primarily on user interface design
- Answer: B

25. What challenge does the Repository architecture face?

- A) It cannot be used in web-based applications
- B) It can become a bottleneck and single point of failure
- C) It is incompatible with modern programming languages
- D) It only allows for sequential data access
- Answer: B

26. Which architecture would you use to ensure that different components can operate independently while sharing data?

- A) Client-server
- B) MVC
- C) Repository
- D) Pipe and filter
- Answer: C

27. What does the client-server architecture inherently support due to its structure?

- A) High coupling between data management and user interface
- B) Distribution of services across a network
- C) Aggregation of all processing in a single server
- D) Use of a single user interface for all system functions
- Answer: B

Based on Chapter 8 on "Software Testing"

1. What is the primary purpose of software testing?

- A) To design software applications.
- B) To demonstrate that a program performs as intended and identify any defects.
- C) To market the software.
- D) To train software users.
- Answer: B

2. What does software testing primarily aim to reveal?

- A) Marketing strategies

- B) The presence of errors

- C) The software development methodology

- D) The programmer's skill level

- Answer: B

3. What is the difference between validation testing and defect testing?

- A) Validation testing is performed by users, while defect testing is performed by developers.

- B) Validation testing aims to demonstrate correctness against the specification, while defect testing aims to find incorrect behavior.

- C) Validation testing checks software usability, while defect testing checks software speed.

- D) There is no difference; both terms describe the same process.

- Answer: B

4. Which type of testing would involve the input-output model to ensure correct behavior?

- A) User testing

- B) System testing

- C) Stress testing

- D) Development testing

- Answer: D

5. Verification and validation are part of which broader process?

- A) Performance management

- B) Software development

- C) Verification and validation process

- D) Quality assurance

- Answer: C

6. Which statement accurately describes 'validation' in software testing?

- A) Ensures the software is built according to the specifications.
- B) Confirms the software meets user requirements and expectations.
- C) Is focused only on internal system characteristics.
- D) None of the above.
- Answer: B

7. What is the role of software inspections?

- A) To execute the software to find runtime errors.
- B) To analyze the static system representation to find defects.
- C) To validate user interfaces.
- D) To monitor software performance.
- Answer: B

8. Which of these is NOT a typical stage in software testing according to Chapter 8?

- A) Development testing
- B) Release testing
- C) Performance testing
- D) Sales testing
- Answer: D

9. What does unit testing focus on?

- A) Testing individual components in isolation.
- B) Testing the interactions between software components.
- C) Testing the complete system for performance.
- D) Testing user interfaces.
- Answer: A

10. How is system testing different from unit testing?

- A) System testing involves only external testers.
- B) System testing integrates components and tests them as a whole.
- C) System testing is not usually automated.
- D) System testing is done without documentation.
- Answer: B

11. What is test-driven development (TDD)?
    - A) A method where testing is done after software development is complete.
    - B) A software development process where tests are written before the software code.
    - C) A regulatory requirement for software testing.
    - D) A type of performance testing.
    - Answer: B

12. What are the advantages of automated testing?
    - A) It eliminates the need for software development.
    - B) It reduces the time needed for repeated test executions.
    - C) It ensures that no defects are ever present.
    - D) It is only used in user testing.
    - Answer: B

13. Which type of testing is used to verify that a system meets the functional requirements?
    - A) Stress testing
    - B) System testing
    - C) Release testing
    - D) Requirements-based testing
    - Answer: D

14. What is a key focus of release testing?
    - A) To perform security audits.

- B) To ensure the software meets the specified requirements before release.
- C) To develop the software according to customer requirements.
- D) To provide training to end users.
- Answer: B

15. Which testing method involves the user in the final stage to validate the software meets their needs?

- A) Unit testing
- B) Acceptance testing
- C) Integration testing
- D) Configuration testing
- Answer: B

16. What is regression testing primarily used for?

- A) To check that new code changes have not adversely affected existing

functionalities.

- B) To test the performance of the software under high loads.
- C) To verify the color scheme of the user interface.
- D) To ensure that the software can be sold in various markets.
- Answer: A

17. What does equivalence partitioning involve in software testing?

- A) Dividing software into various sections based on their sales potential.
- B) Grouping inputs that are processed similarly into equivalence classes.
- C) Partitioning the system into different components based on user roles.
- D) Separating software into different price tiers.
- Answer: B

18. What is the goal of performance testing?

- A) To assess the system's usability.
- B) To validate the system's functional correctness.
- C) To determine the system's responsiveness and stability under load.
- D) To ensure the system is economically viable.
- Answer: C

19. What is the significance of interface testing in software development?

- A) It checks the marketing strategy of the software.
- B) It detects faults in the interaction between interconnected components.
- C) It confirms the software's price point.
- D) It evaluates the software's packaging design.
- Answer: B

20. Which testing technique uses guidelines based on common errors to choose test cases?

- A) Guideline-based testing
- B) Stress testing
- C) User interface testing
- D) Alpha testing
- Answer: A

21. What is the primary purpose of stress testing?

- A) To evaluate the software's features.
- B) To determine how the software behaves under extreme conditions.
- C) To check the software's compliance with international standards.
- D) To assess the software's potential for international sales.
- Answer: B

22. In the context of software testing, what does 'defect testing' aim to achieve?

- A) It ensures the software is defect-free.

- B) It aims to find defects by testing the software under normal and extreme conditions.

- C) It focuses on the aesthetic aspects of the software.

- D) It tests the software's documentation for accuracy.

- Answer: B

23. What is an outcome of successful validation testing?

- A) The software fails to perform as expected.

- B) The software operates according to its functional requirements.

- C) The software's sales figures meet predictions.

- D) The software's source code is optimized.

- Answer: B

24. Which testing approach is described by setting up specific conditions or inputs to observe if the software behaves as expected?

- A) Performance testing

- B) Scenario testing

- C) Random testing

- D) Ad-hoc testing

- Answer: B

25. What distinguishes release testing from system testing?

- A) Release testing is conducted by a separate team and focuses on validation.

- B) Release testing is less comprehensive than system testing.

- C) System testing is performed only by users, not developers.

- D) System testing does not check for defects.

- Answer: A

26. How does test-driven development (TDD) benefit software development?

- A) By ensuring that every piece of code is tested before it is written.
- B) By eliminating the need for testing after development.
- C) By focusing solely on performance optimization.
- D) By reducing the need for project management.
- Answer: A

27. What is the main goal of acceptance testing performed by customers?
    - A) To design the software according to the customer's specifications.
    - B) To decide whether the software is acceptable for deployment.
    - C) To assess the software's market competitiveness.
    - D) To finalize the software's pricing.
    - Answer: B

28. Which type of testing involves testing the system with the intent to make it fail?
    - A) Positive testing
    - B) Negative testing
    - C) Neutral testing
    - D) Non-functional testing
    - Answer: B

Based on Chapter 9 on "Software Evolution,"

1. What is the primary focus of software evolution?
   - A) Redesigning software from scratch.
   - B) Managing and implementing change in existing software systems.
   - C) Decreasing software functionality over time.
   - D) Maintaining hardware components.
   - Answer: B

2. What drives the need for software changes according to the chapter?

- A) Static business environments and unchanged user requirements.
- B) Decreases in software performance only.
- C) Changes in business environments, new requirements, and errors.
- D) New marketing strategies.
- Answer: C

3. Why is software maintenance often more expensive than initial development?

- A) Because it includes redesigning the user interface.
- B) Because it involves frequent system downtime.
- C) Due to the ongoing need to update and adapt existing systems.
- D) Maintenance does not usually exceed development costs.
- Answer: C

4. What is an example of a maintenance activity?

- A) Replacing all hardware components periodically.
- B) Changing a system to operate in a new environment.
- C) Completely rewriting all software applications annually.
- D) Performing external audits.
- Answer: B

5. What is the main challenge in managing legacy systems?

- A) Deciding when to add new features.
- B) Determining whether to replace, maintain, or upgrade them.
- C) Training new users on outdated systems.
- D) Integrating them with modern e-commerce systems.
- Answer: B

6. What does the term 'servicing' refer to in software evolution?

- A) Adding significant new functionalities to the system.
- B) Making essential changes to keep the software operational.
- C) Conducting user training sessions.
- D) Marketing and selling software products.
- Answer: B

7. Which phase comes after 'servicing' in the software evolution process?

- A) Initial development.
- B) Phase-out.
- C) Testing.
- D) Re-engineering.
- Answer: B

8. What is 'program understanding' in the context of change implementation?

- A) Documenting software for marketing purposes.
- B) A phase where developers learn about the program's structure and functionality to implement changes.
- C) A management strategy for promoting software.
- D) The process of user acceptance testing.
- Answer: B

9. What might trigger an urgent change in software?

- A) A planned upgrade of the operating system.
- B) An unforeseen serious fault needing immediate repair.
- C) A routine check for potential enhancements.
- D) A scheduled meeting with stakeholders.
- Answer: B

10. According to the chapter, how do agile methods affect software evolution?

- A) They prevent any need for software evolution.
- B) They make the transition from development to evolution seamless.
- C) They completely eliminate the need for maintenance.
- D) They suggest a complete rewrite of the software periodically.
- Answer: B

11. What is a primary issue during handover from development to maintenance teams?

- A) Transferring hardware between teams.
- B) Differences in methodologies (e.g., agile vs. plan-based) that affect evolution practices.
- C) Legal issues related to software ownership.
- D) Deciding on the software's price.
- Answer: B

12. What is the key concept of Lehman's Laws of Software Evolution?

- A) Software does not need to evolve once it is developed.
- B) Software must evolve to remain useful in its environment.
- C) Software evolution is a minor part of software lifecycle.
- D) All software evolution should be avoided as it is too costly.
- Answer: B

13. Which law reflects the idea that as a program evolves, its complexity increases unless work is done to maintain or reduce it?

- A) Law of Increasing Complexity.
- B) Law of Organizational Stability.
- C) Law of Continuing Change.
- D) Law of Conservation of Familiarity.
- Answer: A

14. What is 'change implementation' in software evolution?

- A) The process of initially building the software.
- B) Designing, implementing, and testing revisions to a system.
- C) The final stage of retiring a software system.
- D) The marketing phase where software changes are promoted.

- Answer: B

15. What is the primary focus of software re-engineering?

- A) Assessing financial viability of software products.
- B) Restructuring or rewriting parts of a system to improve maintainability without changing functionality.
- C) Changing the functional requirements of the system.
- D) Training users on new system features.
- Answer: B

16. Why is refactoring considered preventative maintenance?

- A) It involves changing the software's functionality to meet new market demands.
- B) It focuses on improving software structure to prevent future maintenance issues.
- C) It includes significant changes to the software's user interface.
- D) It is a process of adding new features to meet competitor standards.
- Answer: B

17. What are 'bad smells' in program code, according to the chapter?

- A) Indicators of poor program practices that may hinder maintenance and evolution.
- B) Issues related to hardware that supports the software.
- C) Financial problems related to software development.
- D) Legal issues found during software audits.

- Answer: A

18. What role does the business environment play in software evolution?

- A) It has no impact; software evolution is independent of business changes.
- B) Changes in the business environment often necessitate software changes.
- C) The business environment only affects software pricing.
- D) It is only concerned with how the software is marketed.
- Answer: B

19. What does maintenance to adapt software to a different operating environment involve?

- A) Correcting faults in how the system meets requirements.
- B) Modifying the system so it operates in a new environment like a different OS.
- C) Adding new functionalities to meet user demands.
- D) Phasing out old software systems.
- Answer: B

20. Which of the following best describes maintenance for adding or modifying system functionality?

- A) It involves only cosmetic changes to the software.
- B) It is solely focused on improving software performance.
- C) It includes modifying the system to satisfy new or changed requirements.
- D) It is restricted to changing the software documentation.
- Answer: C

21. What is a key factor that influences maintenance costs?

- A) The age and structure of the program.
- B) The geographical location of the software company.
- C) The color scheme of the software interface.
- D) The number of users who like the software.

- Answer: A

22. How do changes in a software system's environment affect its evolution?

- A) They have no impact as software operates independently of its environment.

- B) Changes in the environment can necessitate software changes to ensure it remains useful.

- C) Environmental changes only affect the software's marketing strategy.

- D) They only affect the hardware on which the software runs.

- Answer: B

23. What is the main concern when managing legacy systems?

- A) Deciding whether to increase the price of the software.

- B) Determining the strategic approach to evolving, maintaining, or replacing the system.

- C) Choosing new colors for the user interface.

- D) Planning new marketing campaigns for old software.

- Answer: B

24. What does the emergency repair process in software evolution typically address?

- A) Routine software updates.

- B) Cosmetic changes to the user interface.

- C) Urgent and critical changes that need immediate implementation.

- D) Long-term strategic planning for software replacement.

- Answer: C

25. What challenge does 'handover' pose in software evolution?

- A) Transferring software from one user to another.

- B) Managing the transition between different development and maintenance teams, especially when they use different methodologies.

- C) Changing the software's name and branding.

- D) Updating the software's documentation to a new format.

- Answer: B

26. According to Lehman's Laws, what happens to a program's structure as it evolves?

- A) It becomes simpler and less detailed.
- B) It tends to become more complex unless efforts are made to reduce this complexity.
- C) It does not change; the structure remains constant.
- D) It periodically resets to its original form.
- Answer: B

27. What does the 'phase-out' stage of a software system signify?

- A) The system is actively developed with new features.
- B) Only critical changes are made; the system is nearing the end of its useful life.
- C) The system undergoes major

re-engineering efforts.

- D) The software is heavily marketed to new users.
- Answer: B

28. What impact does software evolution have on a business's software assets?

- A) No impact, as software assets are static.
- B) Evolution maintains or increases the value of these assets through necessary changes and updates.
- C) Evolution decreases the value by adding unnecessary features.
- D) Software assets are typically phased out during evolution.
- Answer: B

Based on Chapter 14 on "Security Engineering"

1. What is the primary focus of security engineering?

- A) To ensure software usability
- B) To develop systems that resist malicious attacks
- C) To improve software marketing strategies
- D) To reduce software costs
- Answer: B

2. What is the difference between application security and infrastructure security?

- A) Application security focuses on user interface design, infrastructure on backend processes.
- B) Application security is a software engineering problem, infrastructure security is a systems management problem.
- C) Application security deals with external threats only, infrastructure security with internal threats.
- D) There is no difference; both terms describe the same concept.
- Answer: B

3. What does security risk management involve?

- A) Designing user interfaces
- B) Assessing losses from potential attacks and balancing these against security costs
- C) Marketing and selling security software
- D) Training users on new software features
- Answer: B

4. Which of the following is a misuse case scenario for security threats?

- A) An attacker making part of a system unavailable
- B) A user accessing the system with proper credentials
- C) Software updates being installed regularly
- D) Data being stored without encryption
- Answer: A

5. How are security requirements typically developed?

- A) By following a standard set of requirements for all systems
- B) Based on the assessment of security risks
- C) By copying requirements from similar systems
- D) Through user feedback sessions
- Answer: B

6. What is the role of 'life cycle risk assessment' in security engineering?

- A) It is performed only at the beginning of the project.
- B) It assesses risks throughout the development and operational phases.
- C) It only assesses financial risks.
- D) It is unrelated to software development.
- Answer: B

7. What are the main components of a security risk management process?

- A) Preliminary risk assessment, life cycle risk assessment, operational risk assessment
- B) Budgeting, scheduling, resourcing
- C) Code development, testing, deployment
- D) Requirement gathering, design, implementation
- Answer: A

8. Which strategy is used to minimize risks associated with security threats?

- A) Avoiding software updates
- B) Increasing system complexity
- C) Implementing redundant systems and diversified infrastructure
- D) Reducing system functionality
- Answer: C

9. What is meant by 'design for security'?

- A) Designing promotional materials for security products
- B) Designing systems to be secure through architectural decisions and good practices
- C) Designing security software only
- D) Designing for aesthetic appeal
- Answer: B

10. What does 'protection' involve in the context of security engineering?

- A) Legal protection against software piracy
- B) Organizing the system to safeguard critical assets against external attacks
- C) Protecting the physical components of the computer systems
- D) Ensuring that software is protected from competition
- Answer: B

11. What does the distribution strategy in security engineering aim to achieve?

- A) Distribute software copies freely to users
- B) Spread out system assets to minimize attack impact
- C) Share software development tasks globally
- D) Distribute financial risks
- Answer: B

12. What is the goal of application-level protection in security engineering?

- A) To manage user subscriptions
- B) To implement specific protection mechanisms within the application
- C) To protect the application from financial losses
- D) To market the application to a wider audience
- Answer: B

13. What is the primary purpose of attack monitoring and recovery in security engineering?

- A) To ensure all users are trained on security protocols
- B) To monitor the system for unauthorized access and develop strategies for recovery
- C) To monitor employee performance
- D) To monitor the stock market
- Answer: B

14. Which type of security threat involves an attacker making system data unavailable?
    - A) Interception threat
    - B) Modification threat
    - C) Fabrication threat
    - D) Interruption threat
    - Answer: D

15. What is a control

 measure for unauthorized access at the system manager level?

- A) Allowing unrestricted access from any location
- B) Implementing physical security controls at specific locations
- C) Using a single password for all systems
- D) Eliminating the need for authentication
- Answer: B

16. Why is it important to compartmentalize assets in a secure system?
    - A) To ensure all users have access to all information
    - B) To organize the system so users only access necessary information
    - C) To increase system complexity
    - D) To centralize all data
    - Answer: B

17. What role does validation of all inputs play in a secure system?

- A) It guarantees the system will be free of bugs.
- B) It ensures unexpected inputs do not cause problems.
- C) It speeds up the system.
- D) It reduces the cost of the system.
- Answer: B

18. What is the purpose of life-cycle risk analysis in security engineering?

- A) To analyze risks only after deployment
- B) To determine the cost of the system
- C) To identify vulnerabilities arising from design choices
- D) To ensure compliance with legal standards
- Answer: C

19. How can system survivability be enhanced in a secure system design?

- A) By reducing the number of critical services
- B) By ensuring the system can deliver essential services under attack
- C) By completely centralizing system services
- D) By avoiding the use of redundancy
- Answer: B

20. What is the significance of system survivability analysis?

- A) It is only theoretical and has no practical application.
- B) It helps identify strategies to ensure critical services are maintained during attacks.
- C) It focuses solely on financial risks.
- D) It is used to train new employees.
- Answer: B

21. What is an example of a design guideline for secure systems?

- A) Always use the least secure settings for usability.
- B) Base security decisions on an explicit security policy.
- C) Ensure that all systems have a single point of failure.
- D) Avoid logging user actions to increase privacy.
- Answer: B

22. Why should security decisions avoid a single point of failure?
    - A) To ensure that a single failure does not compromise security.
    - B) To make the system easier to use.
    - C) To reduce the cost of the system.
    - D) To comply with international standards.
    - Answer: A

23. What does 'fail securely' imply in the context of system design?
    - A) The system should fail in a way that exposes sensitive information.
    - B) Failures should result in a security lockdown where sensitive data remains protected.
    - C) System failures should be frequent to test user readiness.
    - D) System failures should not be reported to avoid panic.
    - Answer: B

24. How does redundancy contribute to system security?
    - A) By ensuring there are multiple failures before a security breach occurs.
    - B) By keeping multiple copies of data to prevent a single point of failure.
    - C) By making the system more complex and harder to use.
    - D) By increasing the cost of system maintenance.
    - Answer: B

25. What role does diversity play in securing a system?
    - A) It ensures all system components are identical.

- B) It uses varied components to reduce common vulnerabilities.
- C) It makes the system aesthetically pleasing.
- D) It simplifies system management.
- Answer: B

26. What is a key component of secure system deployment?
    - A) Ensuring that security is only considered at the end of the design process.
    - B) Ignoring default settings and configuration vulnerabilities.
    - C) Designing support into the system to minimize deployment errors.
    - D) Focusing on the physical appearance of the hardware.
    - Answer: C

27. Why should security vulnerabilities be addressed during system deployment?
    - A) To increase the system’s marketability.
    - B) Configuration errors made during deployment can introduce security risks.
    - C) To ensure the system uses the most modern technology.
    - D) To comply with aesthetic standards.
    - Answer: B

28. What does survivability mean in the context of security engineering?
    - A) The system's ability to remain operational even when under attack.
    - B) The system's ability to function without any security measures.
    - C) The aesthetic longevity of the system interface.
    - D) The financial profitability of the system.
    - Answer: A

29. What should be considered when designing a system for security?

- A) Only the initial cost of the system.
- B) The potential impact of security decisions on system usability and performance.
- C) The color scheme of the system interface.
- D) The preferences of the system designers.
- Answer: B

30. How can system design practices like logging user actions improve security?

- A) By discouraging irresponsible behavior through monitoring.
- B) By making the system less complex.
- C) By focusing solely on user interface design.
- D) By reducing the number of system features.
- Answer: A