Test Bank Questions
IT242: Software Engineering

Chapter 1

1. Which question no longer concerns the modern software engineer?
    a) Why does computer hardware cost so much?
    b) Why does software take a long time to finish?
    c) Why does it cost so much to develop a piece of software?
    d) Why can't software errors be removed from products prior to delivery?

2. Software is a product and can be manufactured using the same technologies used for other engineering artifacts
    a) True
    b) False

3. Software deteriorates rather than wears out because
    a) Software suffers from exposure to hostile environments
    b) Defects are more likely to arise after software has been used often
    c) Multiple change requests introduce errors in component interactions
    d) Software spare parts become harder to order

4. WebApps are a mixture of print publishing and software development, making their development outside the realm of software engineering practice.
    a) True
    b) False

5. There are no real differences between creating WebApps and MobileApps.
    a) True
    b) False

6. In its simplest form an external computing device may access cloud data services using a web browser.
    a) True
    b) False

7. Product line software development depends on the reuse of existing software components to provide software engineering leverage.
    a) True
    b) False

8. Cloud computing will transform the way in which software is _____.
    a) Delivered
    b) Defined
    c) Deleted
    d) Ordered
9. Legacy systems often evolve for the following reasons:
    a) The software must be adapted to meet the needs of new computing environments or technology.
    b) The software must be enhanced to implement new business requirements.
    c) The software must be extended to make it interoperable with other more modern systems or databases.
    d) All of the above

10. If you want to reduce software deterioration, you'll have to do better _____.
    a) Research
    b) Collaboration
    c) Thinking
    d) Software design

11. Software is both a _____ and a vehicle that delivers a _____.
    a) Product
    b) Process
    c) Design
    d) Concept

12. Every _____ indicates an error in design or in the process through which design was translated into machine executable code.
    a) Hardware failure
    b) Miscalculation
    c) Software failure
    d) Engineering mistake

13. Every software engineer must recognize that
    a) They are the ultimate authority on software design
    b) Change is natural
    c) Clients know little about what they need
    d) Collaboration is unimportant

14. All of these are a category of software except
    a) Collaboration software
    b) Product line software
    c) Application software
    d) Embedded software

15. The implementation of cloud computing requires the development of an architecture that encompasses _____ and _____ services.
    a) Good; bad
    b) Complete; incomplete
    c) Collaborative; combative
    d) Front-end; back-end

16. A software product line is a set of software-intensive systems that share a common, managed _____:
    a) Set of problems
    b) Set of products
    c) Set of features
    d) Set of lines

17. All of the following have contributed to the increasing importance of software over the last half-century except:
    a) Dramatic improvements in hardware performance
    b) Increased wealth in developing nations
    c) Profound changes in computing architecture
    d) Vast increases in memory

Chapter 2

1. Which of the items listed below is not one of the software engineering layers?
    a. Process
    b. Manufacturing
    c. Methods
    d. Tools

2. Software engineering umbrella activities are only applied during the initial phases of software development projects.
    a. True
    b. False

3. Which of these are the 5 generic software engineering framework activities?
    a. communication, planning, modeling, construction, deployment
    b. communication, risk management, measurement, production, reviewing
    c. analysis, designing, programming, debugging, maintenance
    d. analysis, planning, designing, programming, testing

4. Planning ahead for software reuse reduces the cost and increases the value of the systems into which they are incorporated.
    a. True
    b. False

5. The essence of software engineering practice might be described as understand the problem, plan a solution, carry out the plan, and examine the result for accuracy.
    a. True
    b. False

6. In agile process models the only deliverable work product is the working program.
    a. True
    b. False

7. A most software development projects are initiated to try to meet some business need.
    a. True
    b. False

8. In general software only succeeds if its behavior is consistent with the objectives of its designers.
    a. True
    b. False

9. If software has value, it will change over its useful life. For that reason, software must be built to be maintainable.
    a. maintainable
    b. disposable
    c. broken
    d. tested

10. The seven principles Hooker proposes for software engineering as a practice are:
    a. Communication, planning, modeling, construction, deployment, re-design, remodel
    b. The reason it all exists; KISS; maintain the vision; what you produce, others will consume; be open to the future; plan ahead for reuse; think!
    c. The reason it all exists; KISS; maintain the vision; what you produce, others will consume; analysis; design; program
    d. Analysis; planning; maintain the vision; what you produce, others will consume; be open to the future; plan ahead for reuse; think!

11. Polya's essence of problem solving - and the essence of software engineering practice – includes:
    a. Understand the solution, Plan a problem, Carry out a plan, Examine the response
    b. Understand the review, Plan a solution, Carry out the plan, Examine the result for accuracy
    c. Understand the problem, Plan a solution, Carry out the problem, Examine criticism
    d. Understand the problem, Plan a solution, Carry out the plan, Examine the result for accuracy

12. Software process _____ is essential for project success.
    a. communication
    b. planning
    c. adaptation
    d. modeling

13. Both quality and _____ are an outgrowth of good design.
    a. maintainability
    b. communication
    c. review
    d. criticism

14. Understand the _____ before you build a _____.
    a. Communication; deployment
    b. Problem; solution
    c. Design; framework
    d. Question; answer

15. Recognition of software realities is the first step toward formulation of practical _____ for software engineering.
    a. ideas
    b. questions
    c. solutions
    d. reviews

16. Myth or reality? The only deliverable work product for a successful project is the working program.
    a. Myth
    b. Reality

17. Myth or reality? One of the most effective software quality assurance mechanisms can be applied from the inception of a project— the technical review.
    a. Myth
    b. Reality

18. Myth or reality? Software requirements continually change, but change can be easily accommodated because software is flexible.
    a. Myth
    b. Reality

Chapter 6

1. Human aspects of software engineering are not relevant in today's agile process models.
    a) True
    b) False

2. Which of the following is not an important trait of an effective software engineer?
    a) Attentive to detail
    b) Brutally honest
    c) Follows process rule dogmatically
    d) Resilient under pressure

3. Group communication and collaboration are as important as the technical skills of an individual team member to the success of a team.
    a) True
    b) False

4. Teams with diversity in the individual team member skill sets tend to be more effective than teams without this diversity.
    a) True
    b) False

5. Which of the following can contribute to team toxicity?
    a) Frenzied work atmosphere
    b) Inadequate budget
    c) Poorly coordinated software process
    d) Unclear definition of team roles
    e) a,b,d

6. Software engineering team structure is independent of problem complexity and size of the expected software products.
    a) True
    b) False

7. Agile teams are allowed to self-organize and make their own technical decisions.
    a) True
    b) False

8. In XP a metaphor is used as a device to facilitate communications among customers, team members, and managers?
    a) True
    b) False

9. Using an established social media platform negates the need to be concerned about privacy or security.
    a) True
    b) False

10. Use of cloud services can speed up information sharing among software team members?
    a) True
    b) False

11. In collaborative development environments, metrics are used to reward and punish team members.
    a) True
    b) False

12. Which of these factors complicate decision-making by global software teams?
    a) Complexity of problem
    b) Different views of the problem
    c) Law of unintended consequences
    d) Risk associated with decision
    e) All of the above.

Chapter 7

13. Software engineering principles have about a three year half-life.
    a) True
    b) False

14. Which of the following is not one of core principles of software engineering practice?
    a) All design should be as simple as possible, but no simpler.
    b) A software system exists only to provide value to its users.
    c) Pareto principle (20% of any product requires 80% of the effort).
    d) Remember that you produce others will consume

15. Every communication activity should have a facilitator to make sure that the customer is not allowed to dominate the proceedings.
    a) True
    b) False

16. The agile view of iterative customer communication and collaboration is applicable to all software engineering practice.
    a) True
    b) False

17. One reason to involve everyone on the software team in the planning activity is to
    a) adjust the granularity of the plan
    b) control feature creep
    c) get all team members to “sign up” to the plan
    d) understand the problem scope

18. Project plans should not be changed once they are adopted by a team.
    a) True
    b) False

19. Requirements models depict software in which three domains?
    a) architecture, interface, component
    b) cost, risk, schedule
    c) information, function, behavior
    d) None of the above

20. The design model should be traceable to the requirements model?
    a) True
    b) False

21. Teams using agile software practices do not generally create models.
    a) True
    b) False

22. Which of the following is not one of the principles of good coding?
    a) Create unit tests before you begin coding
    b) Create unit tests before you begin coding
    c) Refractor the code after you complete the first coding pass
    d) Write self-documenting code, not program documentation

23. A successful test I ones that discovers at least one as-yet undiscovered error.
    a) True
    b) False
24. Which of the following are valid reasons for collecting customer feedback concerning delivered software?
    a) Allows developers to make changes to the delivered increment
    b) Delivery schedule can be revised to reflect changes Law of unintended consequences
    c) Developers can identify changes to incorporate into next increment
    d) All of the above.
25. Larger programming teams are always more productive than smaller teams.
    a) True
    b) False

Chapter 8

1. Requirements engineering is a generic process that does not vary from one software project to another.
    a) True
    b) False

2. During project inception the intent of the of the tasks are to determine
    a) basic problem understanding
    b) nature of the solution needed
    c) people who want a solution
    d) none of the above
    e) a, b, c

3. Three things that make requirements elicitation difficult are problems of
    a) budgeting
    b) scope
    c) understanding
    d) volatility
    e) b,c,d

4. A stakeholder is anyone who will purchase the completed software system under development.
    a) True
    b) False

5. It is relatively common for different customers to propose conflicting requirements, each arguing that his or her version is the right one.
    a) True
    b) False

6. Which of the following is not one of the context-free questions that would be used during project inception?
    a) What will be the economic benefit from a good solution?
    b) Who is behind the request for work?
    c) Who will pay for the work?
    d) Who will use the solution?

7. Non-functional requirements can be safely ignored in modern software development projects.
    a) True
    b) False

8. In collaborative requirements gathering the facilitator
    a) arranges the meeting place
    b) cannot be a customer
    c) controls the meeting
    d) must be an outsider

9. Which of the following is not one of the requirement classifications used in Quality Function Deployment (QFD)?
    a) exciting
    b) expected
    c) mandatory
    d) normal

10. The work products produced during requirement elicitation will vary depending on the
    a) size of the budget.
    b) size of the product being built.
    c) software process being used.
    d) stakeholders needs.
    e) both a and b

11. User stories are complete descriptions the user needs and include the non-functional requirements for a software increment.
    a) True
    b) False

12. Developers and customers create use-cases to help the software team understand how different classes of end-users will use functions.
    a) True
    b) False

13. Use-case actors are always people, never system devices.
    a) True
    b) False

14. The result of the requirements engineering task is an analysis model that defines which of the following problem domain(s)?
    a) informational
    b) functional
    c) behavioral
    d) all of the above

15. Analysis patterns facilitate the transformation of the analysis model into a design model by suggesting reliable solutions to common problems.
    a) True
    b) False

16. In agile process models requirements engineering and design activities are interleaved.
    a) True
    b) False

17. In win-win negotiation, the customer's needs are met even though the developer's need may not be.
    a) True
    b) False

18. In requirements validation the requirements model is reviewed to ensure its technical feasibility.
    a) True
    b) False

19. The most common reason for software project failure is lack of functionality.
    a) True
    b) False

20. Requirements engineering encompasses seven distinct tasks:
    a) inception, information, elaboration, negotiation, specification, validation, and management
    b) inception, elicitation, function, negotiation, specification, validation, and management
    c) inception, elicitation, elaboration, specification, validation, and management
    d) inception, elicitation, elaboration, negotiation, specification, validation, and management

21. What is quality function deployment?
    a) software engineering term that refers to documented links between software engineering work products
    b) a technique that attempts to translate unspoken customer needs or goals into system requirements
    c) continuity for developers as a project moves from one project phase to another
    d) rows of a traceability matrix

22. All of the following are examples of negotiating guidelines except:
    a) Recognize that it's not a competition.
    b) Map out a strategy.
    c) Do whatever it takes.
    d) Listen actively.

23. Requirements monitoring encompasses all of the following tasks except:
    a) Domain delivery
    b) Distributed debugging
    c) Business activity debugging
    d) Run-time validation

24. The intent of agile requirements engineering is to transfer _____ from stakeholders to the software team.
    a) information
    b) tasks
    c) behavior
    d) ideas

Chapter 9

1. Which of these is not an element of a requirements model?
    a) behavioral elements
    b) class-based elements
    c) data elements
    d) scenario-based elements

2. Which of the following is not an objective for building a requirements model?
    a) define set of software requirements that can be validated
    b) describe customer requirements
    c) develop an abbreviated solution for the problem
    d) establish basis for software design

3. Object-oriented domain analysis is concerned with the identification and specification of reusable capabilities within an application domain.
    a) True
    b) False

4. In structured analysis models focus on the structure of the classes defined for a system along with their interactions.
    a) True
    b) False

5. Creation and refinement of use cases if an important part of scenario-based modeling.
    a) True
    b) False

6. It is important to consider alternative actor interactions when creating a preliminary use case.
    a) True
    b) False

7. Brainstorming is one technique that may be used to derive a complete set of use case exceptions.
    a) True
    b) False

8. In many cases there is no need to create a graphical representation of a usage scenario.
    a) True
    b) False

9. UML activity diagrams are useful in representing which analysis model elements?
    a) Behavioral elements
    b) Class-based elements
    c) Flow-based elements
    d) Scenario-based elements

10. UML swimnlane diagrams allow you to represent the flow of activities by showing the actors having responsibility for creating each data element.
    a) True
    b) False

Chapter 10

11. Which of the following should be considered as candidate objects in a problem space?
    a) Events
    b) People
    c) Structures
    d) All of the above

12. In the grammatical parse of a processing narrative the nouns become object candidates in the analysis model.
    a) True
    b) False

13. Attributes are chosen for an object by examining the problem statement and identifying the entities that appear to be related.
    a) True
    b) False

14. Which of the following is not one of the broad categories used to classify operations?
    a) Computation
    b) Data manipulation
    c) Event monitors
    d) Transformers

15. Collaborators in CRC modeling are those classes needed to fulfill a responsibility on another card.
    a) True
    b) False

16. Which of the following items does not appear on a CRC card?
    a) Class collaborators
    b) Class name
    c) Class reliability
    d) Class responsibilities

17. Class responsibilities are defined by
    a) Its attributes only
    b) Its collaborators
    c) Its operations only
    d) Both its attributes and operations

18. A stereotype is the basis for class reuse in UML modeling.
    a) True
    b) False

19. An analysis package involves the categorization of analysis model elements into useful groupings.
    a) True
    b) False

Chapter 11

20. The behavior modeling is only used in the analysis of real-time systems.
    a) True
    b) False

21. For purposes of behavior modeling an event occurs whenever
    a) a state and process exchange information.
    b) the system an actor exchange information.
    c) two actors exchange information.
    d) two objects exchange information.

22. For purposes of behavior modeling a state is any
    a) consumer or producer of data.
    b) data object hierarchy.
    c) observable mode of behavior.
    d) well defined process.

23. The state transition diagram
    a) depicts relationships between data objects
    b) depicts functions that transform the data flow
    c) indicates how data are transformed by the system
    d) indicates system reactions to external events

24. The UML sequence diagram shows the order in which system events are processed.
    a) True
    b) False

25. Analysis patterns are discovered, they are not explicitly created.
    a) True
    b) False

26. It is not possible to justify the time required for mobile app requirements analysis.
    a) True
    b) False

27. Which is not one of the analysis activities that is used to create a complete analysis model?
    a) Configuration analysis
    b) Content analysis
    c) Functional analysis
    d) Market analysis

28. Content objects are extracted from use cases by examining the scenario description for direct or indirect content references.
    a) True
    b) False

29. What are the elements of a WebApp interaction model?
    a) activity diagrams, sequence diagrams, state diagrams, interface prototype
    b) activity diagrams, collaboration diagrams, sequence diagrams, state diagrams
    c) use-cases, sequence diagrams, state diagrams, interface prototype
    d) use-cases, sequence diagrams, state diagrams, sequence diagrams

30. UML activity diagrams can be used to represent the user observable functionality delivered by the WebApp as well as the operations contained in each analysis class.
    a) True
    b) False

31. Configuration analysis focuses on the architecture of the user's web browsing environment.
    a) True
    b) False