

软件工程 选择题题库

Chapter1: the nature of software.....	2
Chapter 2: Software Engineering.....	2
Chapter 3: Software Process Structure.....	3
Chapter 4: Process Models.....	3
Chapter 5: Agile Development.....	5
Chapter 6: Human Aspects of Software Engineering	6
Chapter 7: Principles that Guide Practice.....	7
Chapter 8: Understanding Requirements.....	8
Chapter 9: Requirements Modeling: Scenario-Based Methods	10
Chapter 10: Requirements Modeling: Class-Based Methods	10
Chapter 11: Requirements Modeling: Behavior, Patterns, and Web/Mobile Apps	11
Chapter 12: Design Concepts	12
Chapter 13: Architectural Design.....	14
Chapter 14: Component-level Design.....	16
Chapter 15: UI Design.....	18
Chapter 16: Pattern-Based Design.....	19
Chapter 17: WebApp Design	21
Chapter 18: MobileApp Design	22
Chapter 19: Quality Concepts.....	23
Chapter 20: Review Techniques.....	25
Chapter 21: Software Quality Assurance.....	26
Chapter 22: Software Testing Strategies.....	27
Chapter 23: Testing Conventional Applications	29
Chapter 24: Testing Object-Oriented Applications	31
Chapter 25: Testing Web Applications.....	32
Chapter 26: Testing Mobile Applications.....	34
Chapter 27: Security Engineering	35
Chapter 28: Formal Modeling and Verification	36
Chapter 29: Software Configuration Management	38
Chapter 30: Product Metrics	40

Chapter 31: Project Management Concepts	42
Chapter 32: Process and Project Metrics.....	43
Chapter 33: Estimation	45
Chapter 34: Scheduling.....	47
Chapter 35: Risk.....	49
Chapter 36: Maintenance and Reengineering.....	50

Caution: 这份题库只是给各位突击用，答案都标明在了原题上。建议没有做过题的先到这边自己过一遍等到对自己的英语感到绝望的时候再来背这份题库效果更佳。

<http://121.42.201.251/se/?switch=7>

前面有些题做了点笔记后面实在是无力做下去了_(:3)反正就当背答案吧.....

Chapter1: the nature of software

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 the reuse of existing software components to provide software engineering leverage.

A. True

B. False

Chapter 2: Software Engineering

1. Which of the items listed below is not one of the software engineering layers?
A. Process
B. Manufacturing
C. Methods
D. Tools
Hint: a "quality" focus
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 (敏捷过程模型) 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

Chapter 3: Software Process Structure

1. Which of the following are recognized process flow types?

A. Concurrent process flow 并行过程流

B. Iterative process flow 迭代过程流

C. Linear process flow 线性过程流

D. Spiral process flow 螺旋过程流?

E. both a and c

(有人看懂了为什么选 E 的话请务必告诉我.....)

2. The communication activity is best handled for small projects using six distinct actions (inception, elicitation, elaboration, negotiation, specification, validation). (启动、需求获取、需求系统、谈判、规格说明和确认)

A. True

B. False

原话: the communication activity might have six distant actions :inception, elicitation, elaboration, negotiation, specification, and validation. Each of these software engineering actions would have many work tasks and a number of distant work products.

3. A good software development team always uses the same task set for every project to insure high quality work products

A. True

B. False

选需。

4. Software processes can be constructed out of pre-existing software patterns to best meet the needs of a software project.

A. True

B. False

5. Which of these are standards for assessing software processes?

A. SEI

B. SPICE

C. ISO 9000

D. ISO 9001

E. both b and d

Chapter 4: Process Models

1. The waterfall model of software development is

A. A reasonable approach when requirements are well defined.

B. A good approach when a working program is required quickly.

C. The best approach to use for projects with large development teams.

D. An old fashioned model that is rarely used any more.

2. The incremental model of software development is

A. A reasonable approach when requirements are well defined.

B. A good approach when a working core product is required quickly.

C. The best approach to use for projects with large development teams.

D. A revolutionary model that is not used for commercial products.

3. Evolutionary software process models

A. Are iterative in nature.

B. Can easily accommodate product requirements changes.

C. Do not generally produce throwaway systems.

D. All of the above.

4. The prototyping model (原型模型) of software development is

A. A reasonable approach when requirements are well defined.

B. A useful approach when a customer cannot define requirements clearly.

C. The best approach to use for projects with large development teams.

D. A risky model that rarely produces a meaningful product.

5. The spiral model of software development

A. Ends with the delivery of the software product.

B. Is more chaotic than the incremental model.

C. Includes project risks evaluation during each iteration.

D. All of the above.

6. The concurrent development model is

A. Another name for concurrent engineering.

B. Defines events that trigger engineering activity state transitions.

C. Only used for development of parallel or distributed systems.

D. Used whenever a large number of change requests are anticipated.

E. Both a and b

7. The component-based development model is

A. Only appropriate for computer hardware design.

B. Not able to support the development of reusable components.

C. Dependent on object technologies for support.

D. Not cost effective by known quantifiable software metrics.

8. The formal methods model of software development makes use of mathematical methods to

A. Define the specification for computer-based systems.

B. Develop defect free computer-based systems.

C. Verify the correctness of computer-based systems.

D. All of the above.

9. Which of these is not one of the phase names defined by the Unified Process model for software development?

A. Inception phase 起始阶段

B. Elaboration phase 细化阶段

C. Construction phase 构建阶段

D. Validation phase

Hint:

Transition phase 转化阶段

Production phase 生产阶段

10. Which of these is not a characteristic of Personal Software Process (个人软件过程)?

A. Emphasizes personal measurement of work product.

B. Practitioner requires careful supervision by the project manager.

C. Individual practitioner is responsible for estimating and scheduling.

D. Practitioner is empowered to control quality of software work products.

11. Which of these are objectives of Team Software Process?

A. Accelerate software process improvement

B. Allow better time management by highly trained professionals

C. Build self-directed software teams

D. Show managers how to reduce costs and sustain quality

E. Both b and c (感觉 A 也对)

Hint:

Show managers how to coach and motivate their teams and how to help them sustain peak performance.

Accelerate software process improvement by making CMM level 5 behavior normal and expected.

Provide improvement guidance to high-maturity organizations.

12. Process technology tools allow software organizations to compress schedules by skipping unimportant activities.

A. True

B. False

13. It is generally accepted that one cannot have weak software processes and create high quality end products.

A. True

B. False

Chapter 5: Agile Development

1. Agility is nothing more than the ability of a project team to respond rapidly to change.

A. True

B. False

2. Which of the following is not necessary to apply agility to a software process?

A. Eliminate the use of project planning and testing

B. Only essential work products are produced

C. Process allows team to streamline tasks

D. Uses incremental product delivery strategy

3. How do you create agile processes to manage unpredictability?

A. Requirements gathering must be conducted very carefully

B. Risk analysis must be conducted before planning takes place

C. Software increments must be delivered in short time periods

D. Software processes must adapt to changes incrementally

E. Both c and d

4. In agile software processes the highest priorities is to satisfy the customer through early and continuous delivery of valuable software.

A. True

B. False

5. In agile development it is more important to build software that meets the customers' needs today than worry about features that might be needed in the future.

A. True

B. False

6. What are the four framework activities found in the Extreme Programming (XP) process model?

A. analysis, design, coding, testing

B. planning, analysis, design, coding

C. planning, analysis, coding, testing

D. planning, design, coding, testing

7. All agile process models conform to a greater or lesser degree to the principles stated in the "Manifesto for Agile Software Development".

A. True

B. False

8. Which is not one of the key questions that is answered by each team member at each daily Scrum meeting?

A. What did you do since the last meeting?

B. What obstacles are you encountering?

C. What obstacles are you encountering?

D. What do you plan to accomplish by the next team meeting?

9. The Dynamic Systems Development Method (DSDM) suggests a philosophy that is

based on the Pareto principle (80% of the application can be delivered in 20% of the time required to build the complete application).

A. True

B. False

10. Agile Modeling (AM) provides guidance to practitioner during which of these software tasks?

A. Analysis

B. Design

C. Coding

D. Testing

E. Both a and b

11. Agile Unified Process uses the classic UP phased activities (inception, elaboration, construction, transition) to help the team visualize the overall process flow.

A. True

B. False

Chapter 6: Human Aspects of Software Engineering

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: Principles that Guide Practice

1. Software engineering principles have about a three year half-life.

A. True

B. False

2. 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

3. Every communication activity should have a facilitator to make sure that the customer

is not allowed to dominate the proceedings.

A. True

B. False

4. The agile view of iterative customer communication and collaboration is applicable to all software engineering practice.

A. True

B. False

5. 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

6. Project plans should not be changed once they are adopted by a team.

A. True

B. False

7. 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

8. The design model should be traceable to the requirements model?

A. True

B. False

9. Teams using agile software practices do not generally create models.

A. True

B. False

10. 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. Refactor the code after you complete the first coding pass**
- D. Write self-documenting code, not program documentation

11. A successful test is one that discovers at least one as-yet undiscovered error.

- A. True**
- B. False

12. 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
- C. Developers can identify changes to incorporate into next increment
- D. All of the above**

13. Larger programming teams are always more productive than smaller teams.

- A. True
- B. False**

Chapter 8: Understanding Requirements

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 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 适用范围、作用域
- C. understanding
- D. volatility

E. b, c, d

4. A stakeholder **(利益相关者)** 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. can not 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. information
- 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

Chapter 9: Requirements Modeling: Scenario-Based Methods

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 is 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 swimlane 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: Requirements Modeling: Class-Based Methods

1. 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

2. In the grammatical parse of a processing narrative the nouns become object candidates in the analysis model.

A. True

B. False

3. Attributes are chosen for an object by examining the problem statement and identifying the entities that appear to be related.

A. True

B. False

4. 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

5. Collaborators in CRC modeling are those classes needed to fulfill a responsibility on another card.

A. True

B. False

6. Which of the following items does not appear on a CRC card?

A. class collaborators

B. class name

C. class reliability

D. class responsibilities

7. Class responsibilities are defined by

A. its attributes only

B. its collaborators

C. its operations only

D. both its attributes and operations

8. A stereotype is the basis for class reuse in UML modeling.

A. True

B. False

9. An analysis package involves the categorization of analysis model elements into useful groupings.

A. True

B. False

Chapter 11: Requirements Modeling: Behavior, Patterns, and

Web/Mobile Apps

1. The behavior modeling is only used in the analysis of real-time systems.

A. True

B. False

2. 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.

3. 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.

4. 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

5. The UML sequence diagram shows the order in which system events are processed.

A. True

B. False

6. Analysis patterns are discovered, they are not explicitly created.

A. True

B. False

7. It is not possible to justify the time required for mobile app requirements analysis.

A. True

B. False

8. 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

9. Content objects are extracted from use cases by examining the scenario description for direct or indirect content references.

A. True

B. False

10. 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

11. 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

12. Configuration analysis focuses on the architecture of the user's web browsing environment.

A. True

B. False

Chapter 12: Design Concepts

1. Which of the following are areas of concern in the design model?

A. architecture

B. data

C. interfaces

D. project scope

E. a, b, c

2. The importance of software design can be summarized in a single word

A. accuracy

B. complexity

C. efficiency

D. quality

3. Which of these are characteristics of a good design?

A. exhibits strong coupling between its modules

B. implements all requirements in the analysis model

C. includes test cases for all components

D. provides a complete picture of the software

E. b and d

4. Which of the following is not a characteristic common to all design methods?

A. configuration management

B. functional component representation

- C. quality assessment guidelines
- D. refinement heuristics

5. What types of abstraction are used in software design?

- A. control
- B. data
- C. environmental
- D. procedural

E. a, b, d

6. Which of the following can be used to represent the architectural design of a piece of software?

- A. Dynamic models
- B. Functional models
- C. Structural models

D. All of the above

7. Design patterns are not applicable to the design of object-oriented software?

- A. True

B. False

8. Since modularity is an important design goal it is not possible to have too many modules in a proposed design.

- A. True

B. False

9. Information hiding makes program maintenance easier by hiding data and procedure from unaffected parts of the program.

A. True

- B. False

10. Cohesion is a qualitative indication of the degree to which a module

- A. can be written more compactly.

B. focuses on just one thing.

- C. is able to complete its function in a timely manner.

- D. is connected to other modules and the outside world.

11. Coupling is a qualitative indication of the degree to which a module

- A. can be written more compactly.

- B. focuses on just one thing.

- C. is able to complete its function in a timely manner.

D. is connected to other modules and the outside world.

12. When using structured design methodologies the process of stepwise refinement is unnecessary.

- A. True

B. False

13. Software designs are refactored to allow the creation of software that is easier to integrate, easier to test, and easier to maintain.

A. True

- B. False

14. Which of the following is not one of the five design class types

- A. Business domain classes

B. Entity classes

- C. Process classes

- D. User interface classes

15. Which design model elements are used to depict a model of information represented from the user's view?

- A. Architectural design elements

- B. Component-level design elements

C. Data design elements

- D. Interface design elements

16. Which design is equivalent to the floor plan of a house?

A. Architectural design

- B. Component-level design
- C. Data design
- D. Interface design

17. Which design model is equivalent to the detailed drawings of the access points and external utilities for a house?

- A. Architectural design
- B. Component-level design
- C. Data design

D. Interface design

18. Which design model is equivalent to a set of detailed drawings for each room in a house?

- A. Architectural design

B. Component-level design

- C. Data design
- D. Interface design

19. The deployment design elements specify the build order for the software components.

- A. True

B. False

Chapter 13: Architectural Design

1. The best representation of system architecture is an operational software prototype.

- A. True

B. False

2. The architectural representations can be an enabler for communication among project

stakeholders.

A. True

- B. False

3. An architectural description is often documented using an architecture template.

- A. True

B. False

4. An architectural decision is often documented using an architecture decision description template.

A. True

- B. False

5. An architectural genre will often dictate the architectural approach that may be used for the structure to be built.

A. True

- B. False

6. An architectural style encompasses which of the following elements?

- A. constraints
- B. set of components
- C. semantic models
- D. syntactic models

E. a, b, c

7. To determine the architectural style or combination of styles that best fits the proposed system, requirements engineering is used to uncover

- A. algorithmic complexity

B. characteristics and constraints

- C. control and data
- D. design patterns

8. Before an architectural pattern can be chosen for use in a specific system it must have

a code implementation to facilitate its reuse.

A. True

B. False

9. The criteria used to assess the quality of an architectural design should be based on system

A. accessibility

B. control

C. data

D. implementation

E. b and c

10. Software architectural considerations often interact with each other and moderate each other.

A. True

B. False

11. Developer notes are not a reliable means of documenting architectural decisions

A. True

B. False

12. During process of modeling the system in context, systems that interact with the target system are represented as

A. Peer-level systems

B. Subordinate systems

C. Superordinate systems

D. Working systems

E. a, b, c

13. Once selected, archetypes always need to be refined further as architectural design proceeds.

A. True

B. False

14. Which of the following is not an example of infrastructure components that may need to be integrated into the software architecture?

A. Communications components

B. Database components

C. Interface components

D. Memory management components

15. In the architecture trade-off analysis method the architectural style should be described using the

A. data flow view

B. module view

C. process view

D. user view

E. a, b, c

16. A useful technique for evaluating the overall complexity of a proposed architecture is to look at the component

A. cohesion

B. flow dependencies

C. sharing dependencies

D. size

E. b and c

17. Software architects need to create consensus among software team members and other stakeholders.

A. True

B. False

18. Pattern-based architectural reviews can be useful for project with short build cycles and volatile requirements.

A. True

B. False

19. Static architectural conformance checking assesses whether or not the source code matches the user visible requirements.

A. True

B. False

20. Architectural design has no role in agile software process models.

A. True

B. False

Chapter 14: Component-level Design

1. In the most general sense a component is a modular building block for computer software.

A. True

B. False

2. In the context of object-oriented software engineering a component contains

A. attributes and operations

B. instances of each class

C. roles for each actor (device or user)

D. set of collaborating classes

3. In traditional software engineering modules must serve in which of the following roles?

A. Control component

B. Infrastructure component

C. Problem domain component

D. All of the above

4. Software engineers always need to create components from scratch in order to meet customer expectations fully.

A. True

B. False

5. Which of the following is not one of the four principles used to guide component-level design?

A. Dependency Inversion Principle

B. Interface Segregation Principle

C. Open-Closed Principle

D. Parsimonious Complexity Principle

6. The use of stereotypes can help identify the nature of components at the detailed design level.

A. True

B. False

7. Classes and components that exhibit functional, layer, or communicational cohesion are relatively easy to implement, test, and maintain.

A. True

B. False

8. Software coupling is a sign of poor architectural design and can always be avoided in every system.

A. True

B. False

9. In component design elaboration requires which of the following elements to be describe in detail?

A. Algorithms

B. Attributes

C. Interfaces

D. Operations

E. b, c, d

10. In component-level design persistent data sources refer to

- A. Component libraries
- B. Databases
- C. Files
- D. All of the above

E. b and c

11. WebApp content design at the component level focuses on content objects and the manner in which they interact.

- A. True

B. False

12. A WebApp functional architecture describes the key functional components and how they interact with each other.

A. True

- B. False

13. Component-level design for mobile apps is not any different from component-based design for Web apps.

A. True

- B. False

14. Which of these constructs is used in structured programming?

- A. branching
- B. condition
- C. repetition
- D. sequence

E. b, c, d

15. In component-based software engineering, the development team examines the requirements to see which are amenable to composition, rather than construction, before beginning detailed design tasks.

A. True

- B. False

16. Which of the following is not one of the major activities of domain engineering?

- A. analysis
- B. construction
- C. dissemination

D. validation

17. Which of the following factors would not be considered during component qualification?

- A. application programming interface (API)
- B. development and integration tools required
- C. exception handling

D. testing equipment required

18. Which of the following is a technique used for component wrapping?

- A. black-box wrapping
- B. clear-box wrapping
- C. gray-box wrapping
- D. white-box wrapping

19. Which of the following is not one of the issues that form a basis for design for reuse?

A. object-oriented programming

- B. program templates
- C. standard data
- D. standard interface protocols

20. In a reuse environment, library queries are often characterized using the _____ element of the 3C Model.

- A. concept
- B. content
- C. context

D. all of the above

Chapter 15: UI Design

1. Which of the following interface design principles does not allow the user to remain in control of the interaction with a computer?

- A. allow interaction to interruptible
- B. allow interaction to be undoable
- C. hide technical internals from casual users

D. only provide one rigidly defined method for accomplishing a task

2. Which of the following interface design principles reduce the user's memory load?

- A. define intuitive shortcuts
- B. disclose information in a progressive fashion
- C. establish meaningful defaults
- D. provide an on-line tutorial

E. a, b, c

3. The reason for reducing the user's memory load is make his or her interaction with the computer quicker to complete.

- A. True

B. False

4. Interface consistency implies that

- A. each application should have its own distinctive look and feel
- B. input mechanisms remain the same throughout the application
- C. navigational methods are context sensitive
- D. visual information is organized according to a design standard

E. b and d

5. If past interactive models have created certain user expectations it is not generally good to make changes to the model.

A. True

B. False

6. Which model depicts the profile of the end users of a computer system?

- A. design model
- B. implementation model
- C. user model
- D. user's model

7. Which model depicts the image of a system that an end user creates in his or her head?

- A. design model
- B. user model
- C. system model

D. system perception

8. Which model depicts the look and feel of the user interface along with all supporting information?

A. implementation model

- B. user model
- C. user's model
- D. system perception

9. Which of these framework activities is not normally associated with the user interface design processes?

A. cost estimation

- B. interface construction
- C. interface validation
- D. user and task analysis

10. Which approach(es) to user task analysis can be useful in user interface design?

- A. have users indicate their preferences on questionnaires
- B. rely on the judgement of experienced programmers
- C. study existing computer-based solutions
- D. observe users performing tasks manually

E. c and d

11. Object-oriented analysis techniques can be used to identify and refine user task objects and actions without any need to refer to the user voice.

A. True

B. False

12. The computer's display capabilities are the primary determinant of the order in which user interface design activities are completed.

A. True

B. False

13. It is sometimes possible that the interface designer is constrained by environmental factors that mitigate against ease of use for many users.

A. True

B. False

14. One means of defining user interface objects and actions is to conduct a grammatical parse of the user scenario.

A. True

B. False

15. Interface design patterns typically include a complete component-level design (design classes, attributes, operations, and interfaces).

A. True

B. False

16. Several common design issues surface for almost every user interface including

A. adaptive user profiles

B. error handling

C. resolution of graphics displays

D. system response time

E. b and d

17. It is more important to capture the user's attention with flashy features than ergonomically sound screen layouts when building a WebApp.

A. True

B. False

18. Several usability measures can be collected while observing users interacting with a computer system including

A. down time for the application

B. number of user errors

C. software reliability

D. time spent looking at help materials

E. b and d

Chapter 16: Pattern-Based Design

1. Which of the following is not one of the elements of a design pattern?

A. context

B. environment

C. problem

D. solution

2. RubberNecking is an example of a classic generative pattern.

A. True

B. False

3. A frame work is a reusable mini-architecture that serves as a foundation which other design patterns can be applied?

A. True

B. False

4. Finding patterns built by others that address design problems is often more difficult

that recognizing patterns in the application to be built.

A. True

B. False

5. A pattern language

A. encompasses a collection of patterns

B. is implemented using hypertext

C. resembles the structure of natural languages

D. None of the above

6. The concepts and techniques discussed for _____ can be used in the conjunction with a pattern-based approach.

A. Architectural design

B. Component-level design

C. User interface design

D. All of the above

7. It is important to reduce the coupling among design patterns so that they can be treated as independent entities.

A. True

B. False

8. Real life design solutions may not always lend themselves to a top-down approach.

A. True

B. False

9. Which of the following problem types are used to label columns in a pattern organizing table?

A. Business

B. Context

C. Database

D. Infrastructure

E. c and d

10. Most mistakes in pattern-based design can be avoided by judicious use of review techniques.

A. True

B. False

11. Before choosing an architectural design pattern it must be assessed for its appropriateness to the application and overall architectural style.

A. True

B. False

12. Unlike architectural patterns, component-level design patterns may be applied to solve subproblems without regard to system context.

A. True

B. False

13. Most user interface design patterns fall within one of ____ categories of patterns.

A. 5

B. 10

C. 25

D. 100

14. WebApp design patterns can be classified by considering which of the dimensions listed below?

A. Aesthetics

B. Design focus

C. Granularity

D. Usability

E. b and c

15. Which of the following are levels of design focus that can be used to categorize WebApp patterns?

A. Behavioral patterns

- B. Functional patterns
- C. Layout patterns
- D. Navigation patterns

E. b and d

16. Which of the levels of granularity that can be used to describe WebApp patterns?

- A. Architectural patterns
- B. Component patterns
- C. Design patterns
- D. Interactions patterns

E. a, b, c

17. Mobile app user interface patterns can be represented as a collection of best of breed screen images.

A. True

B. False

Chapter 17: WebApp Design

1. Which of the following characteristics should not be used to assess the quality of a WebApp?

A. aesthetics

- B. reliability
- C. maintainability
- D. usability

2. Which of the following are design goals for every WebApp?

- A. Simplicity
- B. Consistency
- C. Navigability
- D. Visual appeal

E. All of the above.

3. Which of the following not part of the design pyramid for WebE design?

A. Architectural design

B. Business case design

C. Content design

D. Navigation design

4. With WebApps content is everything, a poorly defined user interface will be quickly overlooked by frequent users.

A. True

B. False

5. Which of these are WebApp interaction mechanisms?

A. Graphic icons

B. Graphic images

C. Navigation menus

D. All of the above

6. Screen layout design has several widely accepted standards based on human factors research.

A. True

B. False

7. Graphic design considers every aspect of the look and feel of a WebApp.

A. True

B. False

8. Content design is conducted by

A. Copywriters and graphic designer

B. Web engineers

C. both a and b

D. none of the above

9. Content objects have both information attributes defined during analysis and implementation specific attributes specified during design.

A. True

B. False

10. Content objects are not normally chunked into Web pages until the implementation activities begin.

A. True

B. False

11. Content architecture and WebApp architecture are pretty much the same thing for many WebApps?

A. True

B. False

12. Which of the following is not one of the content architectural structures used by web engineers?

A. linear

B. grid

C. hierarchical

D. parallel

13. MVC is a three layer architecture that contains a

A. machine, view, content objects

B. model, view, and content objects

C. model, view, and controller

D. machine, view, controller

14. Web navigational design involves creating a semantic navigational unit for each goal associated with each defined user role.

A. True

B. False

15. To allow the user to feel in control of a WebApp, it is a good idea to mix both horizontal and vertical navigation mechanisms on the same page.

A. True

B. False

16. Component level design for WebApps is very similar to component level design for other software delivery environments.

A. True

B. False

17. Which of these is not one of the design activities associated with object-oriented hypermedia design?

A. abstract interface design

B. conceptual design

C. content design

D. navigational design

18. UML does not have any representation schemas that are useful in building WebApp design models.

A. True

B. False

Chapter 18: MobileApp Design

1. MobileApps must be designed take intermittent connectivity outages.

A. True

B. False

2. Modern electronics allow developers to ignore the power demands made by a MobileApp.

A. True

B. False

3. A MobileApp is assessed for usability and accessibility before beginning the next increment begins.

A. True

B. False

4. Which of the following characteristics should not be used to assess the quality of a MobileApp?

A. aesthetics

B. reliability

C. maintainability

D. usability

5. Quality function deployment is not necessary when implementing MobileApp user stories?

A. True

B. False

6. Using highly adaptive contextual interfaces is a good way to deal with device limitations like screen size.

A. True

B. False

7. Which of the following are common MobileApp design mistakes.

A. Inconsistency

B. Interoperability

C. Lean design

D. Overdesigning

E. a and d

8. It is better to multiple short pages than long scrolling forms when implementing mobile device user interfaces.

A. True

B. False

9. Java is the best programming language to use when you want to create portable MobileApps.

A. True

B. False

10. Service computing allows you to avoid the need to integrate service source code into the mobile device client.

A. True

B. False

11. The most important MobileApp architecture decision whether to build a thin or fat mobile client.

A. True

B. False

Chapter 19: Quality Concepts

1. Quality of conformance focuses on the degree to which the implementation of a design meets its requirements and performance goals.

A. True

B. False

2. Which of the following is not one of the attributes of software quality?

A. Adds value for developers and users

B. Effective software process creates infrastructure

C. Removes need to consider performance issues

D. Useful products satisfy stakeholder requirements

3. Product quality can only be assessed by measuring hard quality factors.

A. True

B. False

4. Many software metrics can only be measured indirectly.

A. True

B. False

5. Which of the following are ISO 9126 software quality factors?

A. Functionality

B. Portability

C. Reliability

D. Visual appeal

E. a, b, c

6. Developers need to create a collection of targeted questions to assess each quality factor.

A. True

B. False

7. Software metrics represent direct measures of some manifestation of quality.

A. True

B. False

8. The quality dilemma might be summarized as choosing between building things quickly or building things correctly.

A. True

B. False

9. Good enough software delivers high quality software functions along with specialized functions that contain known bugs.

A. True

B. False

10. Which of the following is likely to be the most expensive cost of quality?

A. Appraisal costs

B. External failure costs

C. Internal failure costs

D. Prevention costs

11. Poor quality leads to software risks that can become serious?

A. True

B. False

12. When a system fails to deliver required functions it is because the customer changes requirements?

A. True

B. False

13. Developers must start focusing on quality during the design phase in order to build secure systems.

A. True

B. False

14. Which of the following management decisions have the potential to impact software quality?

A. Estimation decisions

B. Risk-oriented decisions

C. Scheduling decisions

D. All of the above

15. The project plan should include explicit techniques for _____ and _____ management?

A. change

B. cost

C. error

D. quality

E. a and d

16. Quality control encompasses a set of software engineering actions that help to ensure that each work product meets its quality goals.

A. True

B. False

17. The goal of quality assurance to insure that a software project is error free.

A. True

B. False

Chapter 20: Review Techniques

1. The purpose of software reviews is to uncover errors and defects in work products so they can be removed before moving on to the next phase of development.

A. True

B. False

2. In general the earlier a software defect is discovered and corrected the less costly to the overall project budget.

A. True

B. False

3. Defect amplification models can be used to illustrate the costs associated with using software from its initial deployment to its retirement.

A. True

B. False

4. Review metrics can be used to assess the efficacy of each review activity.

A. True

B. False

5. Defect density can be estimated for any software engineering work product.

A. True

B. False

6. Agile software developers are aware that software reviews always take time without saving any.

A. True

B. False

7. The level of review formality is determined by which of the following?

A. amount of preparation

B. reviewer follow-up

C. size of project budget

D. structure of review

E. a, b, d

8. An informal review may consist of which of the following?

A. casual meeting

B. desk check

C. inspection

D. pair programming

E. a and b

9. Which of the following are objectives for formal technical reviews?

A. allow senior staff members to correct errors

B. assess programmer productivity

C. determining who introduced an error into a program

D. uncover errors in software work products

10. At the end of a formal technical review all attendees can decide to

A. accept the work product without modification

B. modify the work product and continue the review

- C. reject the product due to stylistic discrepancies
- D. reject the product due to severe errors

E. a and d

11. A review summary report answers which three questions?

- A. terminate project, replace producer, request a time extension
- B. what defects were found, what caused defects, who was responsible

C. what was reviewed, who reviewed it, what were the findings

- D. none of the above

12. In any type of technical review, the focus of the review is on the product and not the producer.

A. True

- B. False

13. Sample driven reviews only make sense for very small software development projects.

- A. True

B. False

Chapter 21: Software Quality Assurance

1. Software quality might be defined as conformance to explicitly stated requirements and standards, nothing more and nothing less.

- A. True

B. False

2. People who perform software quality assurance must look at the software from the customer's perspective.

A. True

- B. False

3. The elements of software quality assurance consist of reviews, audits, and testing.

- A. True

B. False

4. Which of these activities is not one of the activities recommended to be performed by an independent SQA group?

- A. prepare SQA plan for the project
- B. review software engineering activities to verify process compliance
- C. report any evidence of noncompliance to senior management

D. serve as the sole test team for any software produced

5. Metrics can be used to indicate the relative strength of a software quality attribute.

A. True

- B. False

6. Attempts to apply mathematical proof to demonstrate that a program conforms to its specifications are doomed to failure.

- A. True

B. False

7. Statistical quality assurance involves

- A. using sampling in place of exhaustive testing of software
- B. surveying customers to find out their opinions about product quality

C. tracing each defect to its underlying cause, isolating the "vital few" causes, and moving to correct them

D. tracing each defect to its underlying causes and using the Pareto principle to correct each problem found

8. Six Sigma methodology defines three core steps.

- A. analyze, improve, control
- B. analyze, design, verify

C. define, measure, analyze

- D. define, measure, control

9. Software reliability problems can almost always be traced to

- A. errors in accuracy
- B. errors in design
- C. errors in implementation
- D. errors in operation

E. b and c

10. Software safety is a quality assurance activity that focuses on hazards that

- A. affect the reliability of a software component

B. may cause an entire system to fail

- C. may result from user input errors
- D. prevent profitable marketing of the final product

11. The ISO quality assurance standard that applies to software engineering is

- A. ISO 9000

B. ISO 9001

- C. ISO 9002
- D. ISO 9003

12. Which of the following is not a section in the standard for SQA plans recommended by IEEE?

A. budget

- B. documentation
- C. reviews and audits
- D. test

Chapter 22: Software Testing Strategies

1. In software quality assurance work there is no difference between software verification and software validation.

- A. True

B. False

2. The best reason for using Independent software test teams is that

- A. software developers do not need to do any testing
- B. strangers will test the software mercilessly
- C. testers do not get involved with the project until testing begins

D. the conflicts of interest between developers and testers is reduced

3. What is the normal order of activities in which traditional software testing is organized?

- A. integration testing, system testing, unit testing, validation testing.
- B. unit testing, validation testing, system testing, integration testing

C. unit testing, integration testing, validation testing, system testing

- D. validation testing, system testing, integration testing, unit testing

4. By collecting software metrics and making use of existing software reliability models it is possible to develop meaningful guidelines for determining when software testing is done.

A. True

- B. False

5. Which of the following strategic issues needs to be addressed in a successful software testing process?

- A. conduct formal technical reviews prior to testing
- B. specify requirements in a quantifiable manner
- C. use independent test teams
- D. wait till code is written prior to writing the test plan

E. a and b

6. Which of the following need to be assessed during unit testing?

- A. algorithmic performance
- B. code stability
- C. error handling
- D. execution paths

E. c and d

7. Units and stubs are not needed for unit testing because the modules are tested independently of one another.

A. True

B. False

8. Top-down integration testing has as its major advantage(s) that

A. low level modules never need testing

B. major decision points are tested early

C. no drivers need to be written

D. no stubs need to be written

E. b and c

9. Bottom-up integration testing has as its major advantage(s) that

A. major decision points are tested early

B. no drivers need to be written

C. no stubs need to be written

D. regression testing is not required

10. Regression testing should be a normal part of integration testing because as a new module is added to the system new

A. control logic is invoked

B. data flow paths are established

C. drivers require testing

D. all of the above

E. a and b

11. Smoke testing might best be described as

A. bulletproofing shrink-wrapped software

B. rolling integration testing

C. testing that hides implementation errors

D. unit testing for small programs

12. When testing object-oriented software it is important to test each class operation separately as part of the unit testing process.

A. True

B. False

13. The OO testing integration strategy involves testing

A. groups of classes that collaborate or communicate in some way

B. single operations as they are added to the evolving class implementation

C. operator programs derived from use-case scenarios

D. none of the above

14. Since many WebApps evolve continuously, the testing process must be ongoing as well.

A. True

B. False

15. Testing MobileApps is not different than testing WebApps. (感觉这辈子英语都白学了)

A. True

B. False

16. The focus of validation testing is to uncover places that a user will be able to observe failure of the software to conform to its requirements.

A. True

B. False

17. Software validation is achieved through a series of tests performed by the user once the software is deployed in his or her work environment.

A. True

B. False

18. Configuration reviews are not needed if regression testing has been rigorously applied during software integration.

A. True

B. False

19. Acceptance tests are normally conducted by the

A. developer

B. end users

C. test team

D. systems engineers

20. Recovery testing is a system test that forces the software to fail in a variety of ways and verifies that software is able to continue execution without interruption.

A. True

B. False

21. Security testing attempts to verify that protection mechanisms built into a system protect it from improper penetration.

A. True

B. False

22. Stress testing examines the pressures placed on the user during system use in extreme environments.

A. True

B. False

23. Performance testing is only important for real-time or embedded systems.

A. True

B. False

24. Debugging is not testing, but always occurs as a consequence of testing.

A. True

B. False

25. Which of the following is an approach to debugging?

A. backtracking

B. brute force

C. cause elimination

D. code restructuring

E. a, b, c

Chapter 23: Testing Conventional Applications

1. With thorough testing it is possible to remove all defects from a program prior to delivery to the customer.

A. True

B. False

2. Which of the following are characteristics of testable software?

A. observability

B. simplicity

C. stability

D. all of the above

3. The testing technique that requires devising test cases to demonstrate that each program function is operational is called

A. black-box testing

B. glass-box testing

C. grey-box testing

D. white-box testing

4. The testing technique that requires devising test cases to exercise the internal logic of a software module is called

A. behavioral testing

B. black-box testing

C. grey-box testing

D. white-box testing

5. What types of errors are missed by black-box testing and can be uncovered by white-box testing?

- A. behavioral errors
- B. logic errors
- C. performance errors
- D. typographical errors

E. b and d

6. Program flow graphs are identical to program flowcharts.

- A. True

B. False

7. The cyclomatic complexity metric provides the designer with information regarding the number of

- A. cycles in the program
- B. errors in the program

C. independent logic paths in the program

- D. statements in the program

8. The cyclomatic complexity of a program can be computed directly from a PDL representation of an algorithm without drawing a program flow graph.

A. True

- B. False

9. Condition testing is a control structure testing technique where the criteria used to design test cases is that they

- A. rely on basis path testing

B. exercise the logical conditions in a program module

- C. select test paths based on the locations and uses of variables
- D. focus on testing the validity of loop constructs

10. Data flow testing is a control structure testing technique where the criteria used to

design test cases is that they

- A. rely on basis path testing
- B. exercise the logical conditions in a program module

C. select test paths based on the locations and uses of variables

- D. focus on testing the validity of loop constructs

11. Loop testing is a control structure testing technique where the criteria used to design test cases is that they

- A. rely basis path testing
- B. exercise the logical conditions in a program module
- C. select test paths based on the locations and uses of variables

D. focus on testing the validity of loop constructs

12. Black-box testing attempts to find errors in which of the following categories

- A. incorrect or missing functions
- B. interface errors
- C. performance errors
- D. none of the above

E. a, b, c

13. Graph-based testing methods can only be used for object-oriented systems

- A. True

B. False

14. Equivalence testing divides the input domain into classes of data from which test cases can be derived to reduce the total number of test cases that must be developed.

A. True

- B. False

15. Boundary value analysis can only be used to do white-box testing.

- A. True

B. False

16. Orthogonal array testing enables the test designer to maximize the coverage of the test cases devised for relatively small input domains.

A. True

B. False

17. Test derived from behavioral class models should be based on the

A. data flow diagram

B. object-relation diagram

C. state transition diagram

D. use-case diagram

18. Documentation does not need to be tested.

A. True

B. False

19. Real-time applications add a new and potentially difficult element to the testing mix

A. performance

B. reliability

C. security

D. time

Chapter 24: Testing Object-Oriented Applications

1. It is not possible to test object-oriented software without including error discovery techniques applied to the system OOA and OOD models.

A. True

B. False

2. The correctness of the OOA and OOD model is accomplished using formal technical reviews by the software quality assurance team.

A. True

B. False

3. The consistency of object-oriented models may be judged by reviewing the CRC card model.

A. True

B. False

4. Test case design for OO software is driven by the algorithmic detail of the individual operations.

A. True

B. False

5. Integration testing of object-oriented software can be accomplished by which of the following testing strategies?

A. Cluster testing

B. Glass-box testing

C. Thread-based testing

D. Use-based testing

E. a, c, d

6. Validation of object-oriented software focuses on user visible actions and outputs from the system.

A. True

B. False

7. Encapsulation of attributes and operations inside objects makes it easy to obtain object state information during testing.

A. True

B. False

8. Use-cases can provide useful input into the design of black-box and state-based tests of OO software.

A. True

B. False

9. Fault-based testing is best reserved for

A. conventional software testing

B. operations and classes that are critical or suspect

C. use-case validation

D. white-box testing of operator algorithms

10. Scenario-based testing

A. concentrates on actor and software interaction

B. misses errors in specifications

C. misses errors in subsystem interactions

D. both a and b

11. Random order tests are conducted to exercise different class instance life histories.

A. True

B. False

12. Which of these techniques is not useful for partition testing at the class level

A. attribute-based partitioning

B. category-based partitioning

C. equivalence class partitioning

D. state-based partitioning

13. Multiple class testing is too complex to be tested using random test cases.

A. True

B. False

14. The state model can be used to derive test cases based on the dynamic behavior of an object-oriented system.

A. True

B. False

Chapter 25: Testing Web Applications

1. Which of the following is not one of the dimensions of quality used to assess a WebApp?

A. Content

B. Maintainability

C. Navigability

D. Usability

2. WebApps require special testing methodologies because WebApp errors have several unique characteristics.

A. True

B. False

3. Since WebnApps evolve continuously, the testing process is an on-going activity, conducted by the Web support staff using regression tests.

A. True

B. False

4. Test planning is not used in WebApp testing.

A. True

B. False

5. As the WebApp architecture is constructed which types of testing are used as integration tests?

A. Component testing

B. Content testing

C. Navigation testing

D. Usability testing

E. both a and c

6. Which of the following is not one of the objectives of WebApp content testing?

A. Find organizational or structure errors

B. Identify linking errors

- C. Uncover semantic errors
- D. Uncover syntactic errors

7. Database testing is very rarely a part of WebApp content testing.

A. True

B. False

8. The overall strategy for interface testing is to uncover errors

- A. in navigation semantics
- B. in overall usability
- C. related to specific interface mechanisms

D. both a and c

9. Which of the following is not a WebApp interface mechanism?

- A. Browser
- B. Cookies

C. Forms

D. Links

10. When testing WebApp interface semantics, each use-case is used as input for the design of a testing sequence.

A. True

B. False

11. Usability tests should be designed and executed by intended users for a given WebApp.

A. True

B. False

12. WebApp compatibility testing is conducted to be sure that the user model for usage scenario matched the user category assigned to a given user.

A. True

B. False

13. Which test case design technique(s) are appropriate for WebApp component-level testing?

- A. Boundary value analysis
- B. Equivalence partitioning
- C. Path testing

D. All of the above

14. The purpose of WebApp navigation syntactic testing is to ensure the correct appearance of each navigation mechanism.

A. True

B. False

15. Both Web engineers and non-technical users conduct navigation semantics testing for WebApps.

A. True

B. False

16. Which of following is not one of the elements that need to be considered when constructing WebApp server-side configuration tests?

A. Browser compatibility

- B. Database software integration
- C. Operating system compatibility
- D. System security measures

17. To design client-side configuration tests each user category is assessed to reduce the number of configuration variables to a manageable number.

A. True

B. False

18. Which of the following is not a testable WebApp security element?

A. Authentication

- B. Encryption
- C. Firewalls

D. Penetration

19. WebApp performance tests are designed to

- A. asses WebApp usability
- B. evaluate page loading times

C. simulate real-world loading situations

- D. test network connectivity

20. Load testing involves determining the input of which 3 variables?

A. N, T, D

- B. N, T, P
- C. T, D, P
- D. N, D, P

21. WebApp stress testing is a continuation load testing.

A. True

- B. False

Chapter 26: Testing Mobile Applications

1. MobileApps require special testing methodologies because of concerns associated using them in diverse network environments.

A. True

- B. False

2. Since MobileApp users are attracted to new technologies they are very tolerant of errors and testing effort can be reduced.

A. True

B. False

3. Designing test cases directly from user stories increase the likelihood of developing effective test cases in a timely manner.

A. True

- B. False

4. Automated testing tools eliminate the need to do regression testing for MobileApps.

A. True

B. False

5. A weighted device platform matrix helps to prioritize test cases.

A. True

B. False

6. Part of the reason for stress testing is to ensure that the MobileApp exhibits graceful degradation on failure.

A. True

- B. False

7. Which of the following are reasons for testing in the wild?

- A. Assessing the impact of production environments
- B. Failing to create test cases
- C. Not understanding user demographics
- D. Testing for variable performance on user devices

E. both a and d

8. When testing the quality of user interaction the focus should be on user visible interaction mechanisms.

A. True

- B. False

9. Which of that following add to the difficulty of testing MobileApp gestures?

- A. Automatic tool use is difficult
- B. Creating functions to simulate events

- C. Screen size variation
- D. Using paper prototypes

E. a, b, c

10. Continuous speech recognition techniques have eliminated the need for key entry in MobileApps.

A. True

B. False

11. Predictive technologies are often used to help speed up virtual keyboard input on mobile devices.

A. True

B. False

12. The ability of a MobileApp to handle alerts without disrupting user workflow must be tested in the production environment?

A. True

B. False

13. The Testing across borders is not necessary each MobileApp is developed for use in a specific country.

A. True

B. False

14. Which of the following are issues that make real-time testing difficult?

- A. Limited device processing capacity
- B. Power limitations on the device
- C. Unique mobile network infrastructures

D. All of the above

15. Device emulators eliminate the need to test MobileApps on actual devices.

A. True

B. False

Chapter 27: Security Engineering

1. When analyzing security requirements focus in system assets with the highest value and greatest exposure.

A. True

B. False

2. It is possible to have a safe system that is not secure.

A. True

B. False

3. Individuals rarely expose their personal information to others on social media networks.

A. True

B. False

4. Wireless networks require the trust and cooperation between nodes that can be exploited by malicious programs?

A. True

B. False

5. Cloud computing has greater levels of security than other web data repositories.

A. True

B. False

6. The security concerns remain an obstacle to implementing the vision implied by the Internet of Things .

A. True

B. False

7. Security and usability requirements are often in conflict with each other.

A. True

B. False

8. Which of following is not one of the elements of a security model?

A. Criminal background checks

B. External interface requirements

C. Rules of operation

D. Security policy objectives

9. Security metrics and measures need to assess which of these properties?

A. Dependability

B. Survivability

C. Trustworthiness

D. All of the above

10. Security correctness checks should be included which of the following activities?

A. Audits

B. Deployment

C. Inspections

D. Testing

E. a, b, c

11. Which is not one of the elements of a security case?

A. Arguments

B. Bug reports

C. Claims

D. Evidence

12. Security assurance and risk identification must be included in the schedule and budget if they are to be taken seriously.

A. True

B. False

13. Threat analysis is not needed for conventional software applications.

A. True

B. False

14. An incident response plan spells out the actions to be carried out by each stakeholder in response to specific attacks.

A. True

B. False

Chapter 28: Formal Modeling and Verification

1. The cleanroom strategy is based on the _____ software process model.

A. evolutionary

B. incremental

C. revolutionary

D. spiral

2. The cleanroom strategy relies on

A. exhaustive testing

B. extensive unit testing of all modules

C. tests that exercise the software as it is really used

D. white box testing strategies

3. Use of formal program correctness proofs as part of the cleanroom process eliminates the need do any testing for software defects.

A. True

B. False

4. In cleanroom software engineering a "box" encapsulates some system aspect at a particular level of detail.

A. True

B. False

5. This box specification describes an abstraction, stimuli, and response.

A. black box

- B. clear box
- C. state box
- D. white box

6. This box specification describes the architectural design for some system component.

- A. black box
- B. clear box
- C. state box
- D. white box

7. This box specification is closely aligned with procedural design and structured programming.

- A. black box
- B. clear box
- C. state box
- D. white box

8. In cleanroom software engineering the structured programming approach is used to

- A. refine data design
- B. refine function design
- C. refine usage test cases

D. both a and b

9. By using only structured programming constructs as you create a procedural design, you make the work of proving design correctness much easier.

- A. True
- B. False

10. Which of the following is not an advantage of using rigorous correctness verification of each refinement of the clear box design?

A. improves performance of code

- B. produces better code than unit testing
- C. reduces verification effort
- D. results in near zero defect levels

11. Statistical use testing relies on probability distributions based on

- A. mixture of control structures used in the program
- B. order in which the module execute
- C. the way software will actually be used
- D. user interface design standards

12. Certification of an increment is complete once it has passed the formal verification process.

- A. True
- B. False

13. Which of the following models is part of the cleanroom certification process?

- A. component model
- B. sampling model
- C. both a and b
- D. none of the above

14. A data invariant is a set of conditions that are true during the execution of any function.

- A. True
- B. False

15. In some formal languages, stored data that the system accesses and alters is called a(n)

- A. attribute
- B. data structure
- C. state
- D. variant

16. In formal methods work, an action that reads or writes data to a state is called a(n)

- A. actor
- B. event
- C. invariant
- D. operation**

17. What defines the circumstances in which a particular operation is valid?

- A. data invariant
- B. precondition**
- C. postcondition
- D. state

18. Using formal methods eliminates the need to write natural language commentary in the specification document.

- A. True
- B. False**

Chapter 29: Software Configuration Management

1. Which of these are valid software configuration items?

- A. case tools
- B. documentation
- C. executable programs
- D. test data

E. All of the above.

2. Which of the following is not considered one of the four important elements that should exist when a configuration management system is developed?

- A. component elements
- B. human elements
- C. process elements
- D. validation elements**

3. Once a software engineering work product becomes a baseline it cannot be changed again.

- A. True
- B. False**

4. Which configuration objects would not typically be found in the project database?

- A. design specification
- B. marketing data
- C. organizational structure description
- D. test plans

E. b and c

5. Modern software engineering practices usually attempt to maintain SCI's in a project database or repository.

- A. True**
- B. False

6. A data repository meta model is used to determine how

- A. information is stored in the repository
- B. well data integrity can be maintained
- C. easily the existing model can be extended

D. all of the above

7. Many data repository requirements are the same as those for a typical database application.

- A. True**
- B. False

8. The ability to track relationships and changes to configuration objects is one of the most important features of the SCM repository.

- A. True**
- B. False

9. Which of the following tasks is not part of software configuration management?

- A. change control
- B. reporting

C. statistical quality control

- D. version control

10. A basic configuration object is a _____ created by a software engineer during some phase of the software development process.

- A. program data structure
- B. hardware driver

C. unit of information

- D. all of the above

11. Version control systems establish a change set as part of their primary functionality.

- A. True

B. False

12. Change control is not necessary if a development group is making use of an automated project database tool.

- A. True

B. False

13. When software configuration management is a formal activity the software configuration audit is conducted by the

- A. development team

B. quality assurance group

- C. senior managers
- D. testing specialists

14. The primary purpose of configuration status reporting is to

- A. allow revision of project schedule and cost estimates by project managers
- B. evaluate the performance of software developers and organizations

C. make sure that change information is communicated to all affected parties

- D. none of the above

15. Configuration issues that need to be considered when developing Web and Mobile Apps include:

- A. content
- B. cost
- C. people
- D. politics

E. a, b, c

16. Web and Mobile App configuration objects can be managed in much the same way as conventional software configuration objects except for:

A. content items

- B. functional items
- C. graphic items
- D. user items

17. Content management establishes a process by which Web content is rendered on the user's display screen.

- A. True

B. False

18. Change management for Web and Mobile Apps is best handled in agile manner.

A. True

- B. False

19. One reason that version control is difficult for WebApps is that in an uncontrolled environment, you can have multiple authors making changes to the same files from multiple locations without any realizing it.

A. True

- B. False

20. Requiring developers to check Web configuration items in and out and sending affected stakeholders e-mail messages automatically are good ways to deal with configuration auditing and reporting for WebApps.

A. True

B. False

Chapter 30: Product Metrics

1. Most technical software metrics described in this chapter represent indirect measures software attributes that are useful in the quantitative assessment of software quality.

A. True

B. False

2. Which these are reasons for using technical product measures during software development?

A. large body of scientific evidence supports their use

B. provides software engineers with an objective mechanism for assessing software quality

C. they allow all quality software quality information to be expressed unambiguously as a single number

D. all of the above

3. Which measurement activity is missing from the list below?

A. design

B. feedback

C. measurement

D. quantification

4. The Goal/Question/Metric (GQM) paradigm was developed as a technique for assigning blame for software failures.

A. True

B. False

5. One of the most important attributes for a software product metric is that it should be

A. easy to compute

B. qualitative in nature

C. reliable over time

D. widely applicable

6. In many cases metrics for one model may be used in later software engineering activities (e.g. design metrics may be used in test planning).

A. True

B. False

7. The function point metric is an example of metric that can be used to assist with technical decision-making based on the analysis model information, without making use of historical project data.

A. True

B. False

8. The specification metrics proposed by Davis address which two characteristics of the software requirements?

A. functionality and performance

B. performance and completeness

C. specificity and completeness

D. specificity and functionality

9. Architectural design metrics focus on

A. architectural structure

B. data structural relationships

C. internal module complexity

D. module effectiveness

E. a and d

10. Which of the following is not a measurable characteristic of an object-oriented design?

A. completeness

B. efficiency

C. size

D. volatility

11. The depth of inheritance tree (DIT) metric can give an OO software designer a reading on the

A. attributes required for each class

B. completion time required for system implementation

C. complexity of the class hierarchy

D. level of object reusability achieved

12. Because the class is the dominant unit in OO systems there is no call for the definition of class-oriented metrics.

A. True

B. False

13. If you encounter a class with a large responsibility (large class size or CS value) you should consider

A. making it a base class

B. making it a subclass

C. partitioning the class

D. starting a new class hierarchy

14. Component-level metrics include measures of

A. complexity

B. coupling

C. module cohesion

D. performance

E. a, b, c

15. Because the class is the dominant unit in OO systems very few metrics have been proposed for operations that reside within a class.

A. True

B. False

16. Interface metrics are use to assess the complexity of the module's input and output relationships with external devices.

A. True

B. False

17. Most WebApps can be easily characterized by judicious use of widely recognized suites of software metrics?

A. True

B. False

18. Halstead's source code metrics are based on the number of

A. modules in the program

B. operands in the program

C. operators in the program

D. volume elements in the program

E. b and c

19. Software testing metrics fall into two broad categories

A. metrics that focus on defect removal effectiveness

B. metrics that focus on test coverage

C. metrics that estimate the duration of the testing process

D. metrics that predict the number of test cases required

E. b and d

20. The IEEE software maturity index (SMI) is used to provide a measure of the

A. maintainability of a software product based on its availability

B. relative age of a software product being considered for retirement

C. reliability of a software product following regression testing

D. stability of a software product as it is modified during maintenance

Chapter 31: Project Management Concepts

1. Effective software project management focuses on

- A. people, performance, payoff, product
- B. people, product, performance, process

C. people, product, process, project

- D. people, process, payoff, product

2. Organizations that achieve high levels of maturity in people management have a higher likelihood of implementing effective software engineering processes.

A. True

- B. False

3. The first step in project planning is to

- A. determine the budget.
- B. select a team organizational model.
- C. determine the project constraints.

D. establish the objectives and scope.

4. Process framework activities are populated with

- A. milestones
- B. work products
- C. QA points

D. all of the above

5. Project management is less important for modern software development since most projects are successful and completed on time.

- A. True

B. False

6. Which of the following is not considered a stakeholder in the software process?

- A. customers

- B. end-users

- C. project managers

D. sales people

7. The best person to hire as a project team leader is the most competent software engineering practitioner available.

- A. True

B. False

8. The best project team organizational model to use when tackling extremely complex problems is the

- A. closed paradigm

B. open paradigm

- C. random paradigm

- D. synchronous paradigm

9. Which factors should be considered in choosing the organizational structure for a software team?

- A. degree of communication desired
- B. predicted size of the resulting program
- C. rigidity of the delivery date
- D. size of the project budget

E. a, b, c

10. One of the best ways to avoid frustration during the software development process is to

A. give team members more control over process and technical decisions.

- B. give team members less control over process and technical decisions.
- C. hide bad news from the project team members until things improve.
- D. reward programmers based on their productivity.

11. Small agile teams have no place in modern software development.

- A. True

B. False

12. Which of these software characteristics is not a factor contributing to project coordination difficulties?

A. interoperability

B. performance

C. scale

D. uncertainty

13. Which of these software characteristics are used to determine the scope of a software project?

A. context, lines of code, function

B. context, function, communication requirements

C. information objectives, function, performance

D. communications requirements, performance, information objectives

14. The major areas of problem decomposition during the project scoping activity are the

A. customer workflow

B. functionality to be delivered

C. process used to deliver functionality

D. software process model

E. b and c

15. Product and process decomposition occurs simultaneously as the project plan evolves.

A. True

B. False

16. When can selected common process framework activities be omitted during process decomposition?

A. when the project is extremely small in size

B. any time the software is mission critical

C. rapid prototyping does not require their use

D. never the activities are invariant

17. How does a software project manager need to act to minimize the risk of software failure?

A. double the project team size

B. request a large budget

C. start on the right foot

D. track progress

E. c and d

18. The W5HH principle contains which of the following questions?

A. Why is the system being developed?

B. What will be done by whom?

C. Where are they organizationally located?

D. How much of each resource is required?

E. a, c d

19. Which of these are critical practices for performance-based project management?

A. assessing product usability

B. defect tracking against quality targets

C. empirical cost estimation

D. formal risk management

E. b, c, d

Chapter 32: Process and Project Metrics

1. Which of these are valid reasons for measuring software processes, products, and resources?

A. to characterize them

B. to evaluate them

C. to price them

D. to improve them

E. a, b, d

2. Process indicators enable a software project manager to

- A. assess the status of an on-going project
- B. track potential risks
- C. adjust work flow or tasks

D. none of the above

3. Public metrics are used

- A. to evaluate the performance of software development teams.
- B. to appraise the performance of individual team members.
- C. to make strategic changes to the software process.
- D. to make tactical changes during a software project.

E. c and d

4. Which of the following items are not measured by software project metrics?

- A. inputs
- B. markets
- C. outputs
- D. results

5. Software quality and functionality must be measured indirectly.

- A. True
- B. False

6. Which of following are advantages of using LOC (lines of code) as a size-oriented metric?

A. LOC is easily computed.

- B. LOC is a language dependent measure.
- C. LOC is a language independent measure.
- D. LOC can be computed before a design is completed.

7. Which of the following are advantages of using function points (FP) as a measure of the functionality delivered by a software application?

- A. FP is easily computed.

B. FP is a language dependent measure.

C. FP is a language independent measure.

D. FP can be computed before a design is completed.

E. c and d

8. There is no need to reconcile LOC and FP measures since each is meaningful in its own right as a project measure.

A. True

B. False

9. Object-Oriented project measures may be combined with historical project data to provide metrics that aid in project estimation.

A. True

B. False

10. Use-Case oriented metrics are computed directly from UML diagrams they are often used as normalization measures.

A. True

B. False

11. Which of the following is not a measure that can be collected from a Web application project?

A. Customization index

B. Number of dynamic objects

C. Number of internal page links

D. Number of static web pages

12. Which of the following software quality factors is most likely to be affected by radical changes to computing architectures?

A. operation

B. transition

C. revision

D. none of the above

13. Which of the following provide useful measures of software quality?

- A. correctness, performance, integrity, usability
- B. reliability, maintainability, integrity, sales
- C. correctness, maintainability, size, satisfaction

D. correctness, maintainability, integrity, usability

14. A software quality metric that can be used at both the process and project levels is defect removal efficiency (DRE).

A. True

B. False

15. Why is it important to measure the process of software engineering and software it produces?

- A. It is really not necessary unless the project is extremely complex.
- B. To determine costs and allow a profit margin to be set.

C. To determine whether a software group is improving or not.

D. To make software engineering more like other engineering processes.

16. To be an effective aid in process improvement the baseline data used must be:

- A. based on reasonable guestimates from past projects
- B. measured consistently across projects
- C. drawn from similar projects
- D. based on all previously completed projects

E. b and c

17. Baseline data must be collected in an on-going manner and cannot be computed by formal study of historical project data.

A. True

B. False

18. Small software organizations are not likely to see any economic return from establishing software metrics program.

A. True

B. False

19. The software metrics chosen by an organization are driven by the business or technical goals an organization wishes to accomplish.

A. True

B. False

Chapter 33: Estimation

1. Since project estimates are not completely reliable, they can be ignored once a software development project begins.

A. True

B. False

2. The objective of software project planing is to

- A. convince the customer that a project is feasible.
- B. make use of historical project data.

C. enable a manager to make reasonable estimates of cost and schedule.

D. determine the probable profit margin prior to bidding on a project.

3. The project scope is defined as a means of bounding the system

- A. functionality
- B. performance
- C. costs
- D. schedule

E. a and b

4. Software feasibility is based on which of the following

- A. business and marketing concerns
- B. scope, constraints, market

C. technology, finance, time, resources

D. technical prowess of the developers

5. The number of people required for a software project is determined

A. after an estimate of the development effort is made.

B. by the size of the project budget.

C. from an assessment of the technical complexity of the system.

D. all of the above

6. Reusable software components must be

A. catalogued for easy reference.

B. standardized for easy application.

C. validated for easy integration.

D. all of the above

7. The software engineering environment (SEE) consists of which of the following?

A. customers

B. developers

C. hardware platforms

D. software tools

E. c and d

8. Software project estimation techniques can be broadly classified under which of the following headings?

A. automated processes

B. decomposition techniques

C. empirical models

D. regression models

E. b and c

9. The size estimate for a software product to be built must be based on a direct measure like LOC.

A. True

B. False

10. Problem-based estimation is based on problem decomposition which focuses on

A. information domain values

B. project schedule

C. software functions

D. process activities

E. a and c

11. LOC-based estimation techniques require problem decomposition based on

A. information domain values

B. project schedule

C. software functions

D. process activities

12. FP-based estimation techniques require problem decomposition based on

A. information domain values

B. project schedule

C. software functions

D. process activities

13. Process-based estimation techniques require problem decomposition based on

A. information domain values

B. project schedule

C. software functions

D. process activities

E. c and d

14. Unlike a LOC or function point each person's "use-case" is exactly the same size.

A. True

B. False

15. When agreement between estimates is poor the cause may often be traced to inadequately defined project scope or inappropriate productivity data.

A. True

B. False

16. Empirical estimation models are typically based on

A. expert judgement based on past project experiences

B. refinement of expected value estimation

C. regression models derived from historical project data

D. trial and error determination of the parameters and coefficients

17. COCOMO II is an example of a suite of modern empirical estimation models that require sizing information expressed as:

A. function points

B. lines of code

C. object points

D. any of the above

18. Putnam's software equation is a dynamic empirical model that has two independent parameters: a size estimate and an indication of project duration in calendar months or years.

A. True

B. False

19. Function points are of no use in developing estimates for object-oriented software.

A. True

B. False

20. In agile software development estimation techniques focus on the time required to complete each

A. increment

B. scenario

C. task

D. use-case

21. It is possible to use a modified function point technique to develop estimates for Web applications.

A. True

B. False

22. Using a statistical technique like decision tree analysis can provide some assistance in sorting out the true costs associated with the make-buy decision.

A. True

B. False

23. Outsourcing always provides a simple means of acquiring software at lower cost than onsite development of the same product.

A. True

B. False

Chapter 34: Scheduling

1. Software projects are inevitably late and there is nothing that can explain why.

A. True

B. False

2. It is unethical to undertake a project that you know in advance cannot be completed by the customer's deadline, unless you inform the customer of the risk and establish a project plan that can deliver the needed system incrementally.

A. True

B. False

3. Which of the following is not one of the guiding principles of software project scheduling:

A. compartmentalization

B. market assessment

C. time allocation

D. effort validation

4. Doubling the size of your software project team is guaranteed to cut project completion time in half.

A. True

B. False

5. The software equation can be used to show that by extending the project deadline slightly

A. fewer people are required

B. you are guaranteed to meet the deadline

C. more lines of code can be produced

D. none of the above

6. The 40-20-40 rule suggests that the least of amount of development effort be spent on

A. estimation and planning

B. analysis and design

C. coding

D. testing

7. A task set is a collection of

A. engineering work tasks, milestones, deliverables

B. task assignments, cost estimates, metrics

C. milestones, deliverables, metrics

D. responsibilities, milestones, documents

8. The task (activity) network is a useful mechanism for

A. computing the overall effort estimate

B. detecting intertask dependencies

C. determining the critical path

D. specifying the task set to the customer

E. b and c

9. Tasks that lie on the critical path in a task network may be completed in any order as long as the project is on schedule.

A. True

B. False

10. Two tools for computing critical path and project completion times from activity networks are

A. CPM

B. DRE

C. FP

D. PERT

E. a and d

11. Timeline charts assist project managers in determining what tasks will be conducted at a given point in time.

A. True

B. False

12. The best indicator of progress on a software project is the completion

A. of a defined engineering activity task

B. of a successful budget review meeting on time

C. and successful review of a defined software work product

D. and successful acceptance of project prototype by the customer

13. Since iterative process model work best for object-oriented projects it is impossible to determine whether an increment will be completed on time or not.

A. True

B. False

14. WebApp projects only require the creation of a macro schedule.

A. True

B. False

15. The purpose of earned value analysis is to

- A. determine how to compensate developers based on their productivity
- B. provide a quantitative means of assessing software project progress**
- C. provide a qualitative means of assessing software project progress
- D. set the price point for a software product based on development effort

16. Earned value analysis is a technique that allows managers to take corrective action before a project crisis develops.

A. True

B. False

Chapter 35: Risk

1. Proactive risk management is sometimes described as fire fighting.

A. True

B. False

2. Software risk always involves two characteristics

- A. fire fighting and crisis management
- B. known and unknown risks

C. uncertainty and loss

D. staffing and budget

3. Three categories of risks are

A. business risks, personnel risks, budget risks

B. project risks, technical risks, business risks

C. planning risks, technical risks, personnel risks

D. management risks, technical risks, design risks

4. Generic risks require far more attention than product-specific risks.

A. True

B. False

5. A risk item checklist would contain known and predictable risks from which of these categories?

- A. product size
- B. development environment
- C. staff size
- D. process definition

E. all of the above

6. Questions that should be asked to assess the overall project risk include:

- A. Have top managers formally committed to support the project?
- B. Are end-users committed to the project and proposed system being built?
- C. Are requirements fully understood by development team and customers?
- D. Does the proposed budget have time allocated for marketing?

E. a, b, c

7. Software risk impact assessment should focus on consequences affecting

- A. planning, resources, cost, schedule
- B. marketability, cost, personnel
- C. business, technology, process

D. performance, support, cost, schedule

8. Risk projection attempts to rate each risk in two ways

- A. likelihood and cost
- B. likelihood and impact

C. likelihood and consequences

D. likelihood and exposure

9. Risk tables are sorted by

A. probability and cost

B. probability and impact

C. probability and consequences

D. probability and exposure

10. Individual team members can make their own estimate for a risk probability and then develop a consensus value.

A. True

B. False

11. Which factors affect the probable consequences likely if a risk does occur?

A. risk cost

B. risk timing

C. risk scope

D. risk resources

E. b and c

12. The reason for refining risks is to break them into smaller units having different consequences.

A. True

B. False

13. Effective risk management plan needs to address which of these issues?

A. risk avoidance

B. risk monitoring

C. contingency planning

D. all of the above

14. Risk monitoring involves watching the risk indicators defined for the project and not determining the effectiveness of the risk mitigation steps themselves.

A. True

B. False

15. Hazard analysis focuses on the identification and assessment of potential hazards that can cause

A. project termination

B. schedule slippage

C. cost overruns

D. an entire system to fail

16. Risk information sheets (RIS) are never an acceptable substitute for a full risk mitigation, monitoring, and management (RMMM) plan.

A. True

B. False

Chapter 36: Maintenance and Reengineering

1. How much effort is typically expended by a software organization on software maintenance?

A. 20 percent

B. 40 percent

C. 60 percent

D. 80 percent

2. Software supportability is not concerned with either the provision of hardware or infrastructure.

A. True

B. False

3. Business process reengineering is often accompanied by software reengineering.

A. True

B. False

4. Which of the following is not an example of a business process?

A. designing a new product

B. hiring an employee

C. purchasing services

D. testing software

5. Business process reengineering does not have a start or end, it is an evolutionary process.

A. True

B. False

6. Which of the following activities is not part of the software reengineering process model?

A. forward engineering

B. inventory analysis

C. prototyping

D. reverse engineering

7. Software reengineering process model includes restructuring activities for which of the following work items?

A. code

B. documentation

C. data

D. all of the above

8. Which of the following is not an issue to consider when reverse engineering?

A. abstraction level

B. completeness

C. connectivity

D. directionality

9. Reverse engineering of data focuses on

A. database structures

B. internal data structures

C. both a and b

D. none of the above

10. The first reverse engineering activity involves seeking to understand

A. data

B. processing

C. user interfaces

D. none of the above

11. Reverse engineering should proceed the reengineering of any user interface.

A. True

B. False

12. Which of these benefits can be achieved when software is restructured?

A. higher quality programs

B. reduced maintenance effort

C. software easier to test

D. all of the above

13. Code restructuring is a good example of software reengineering.

A. True

B. False

14. Which of these is not an example of data restructuring?

A. data analysis

B. data name rationalization

C. data record standardization

D. none of the above

15. Forward engineering is not necessary if an existing software product is producing the correct output.

A. True

B. False

16. Reengineering client/server systems begins with a thorough analysis of the business environment that encompasses the existing computing system.

A. True

B. False

17. The only time reengineering enters into work with a legacy system is when its components will be implemented as objects.

A. True

B. False

18. The cost benefits derived from reengineering are realized largely due to decreased maintenance and support costs for the new software product.

A. True

B. False