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

Today the increased power of the personal computer has brought about an abandonment of the practice of team development of software.

- A) True
- B) False

3

Software is a product and can be manufactured using the same technologies used for other engineering artifacts.

- A) True
- B) False

4

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

5

Most software continues to be custom built because

- A) Component reuse is common in the software world.
- B) Reusable components are too expensive to use.
- C) Software is easier to build without using someone else's components.
- D) Off-the-shelf software components are unavailable in many application domains.

6

The nature of software applications can be characterized by their information

- A) complexity

B content)

C determinacy)

D both b and c)

7

Modern software applications are so complex that it is hard to develop mutually exclusive category names.

A True)

B False)

8

The so called "new economy" that gripped commerce and finance during the 1990s died and no longer influences decisions made by businesses and software engineers.

A True)

B False)

9

The functionality of most computer systems does not need to be enhanced the lifetime of the system.

A True)

B False)

10

Change cannot be easily accommodated in most software systems, unless the system was designed with change in mind.

A True)

B False)

11

Most software development projects are initiated to try to meet some business need.

A True)

B False)

12

In general software only succeeds if its behavior is consistent with the objectives of its designers.

A True)

B False)

4 CORRECT

WebApps are a mixture of print publishing and software development, making their development outside the realm of software engineering practice.

A) True

✓ B) False

Feedback: (Section 1.2)

5 CORRECT

There are no real differences between creating WebApps and MobileApps.

A) True

✓ B) False

Feedback: (Section 1.2)

6 CORRECT

In its simplest form an external computing device may access cloud data services using a web browser.

✓ A) True

B) False

Feedback: (Section 1.2)

7 CORRECT

Product line software development depends the reuse of existing software components to provide software engineering leverage.

✓ A) True

B) False

the correct answer for each question is indicated by a ✓.

1 CORRECT

Which of the items listed below is not one of the software engineering layers?

A) Process

✓ B) Manufacturing

C) Methods

D) Tools

Feedback: (Section 2.1)

2 CORRECT

Software engineering umbrella activities are only applied during the initial phases of software development projects.

A) True

✓ B) False

Feedback: (Section 2.2)

3 CORRECT

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

Feedback: (Section 2.2)

4 CORRECT

Planning ahead for software reuse reduces the cost and increases the value of the systems into which they are incorporated.

✓ **A)** True

B) False

Feedback: (Section 2.3)

5 CORRECT

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

Feedback: (Section 2.3)

6 CORRECT

In agile process models the only deliverable work product is the working program.

A) True

✓ **B)** False

Feedback: (Section 2.4)

7 CORRECT

A most software development projects are initiated to try to meet some business need.

✓ **A)** True

B) False

Feedback: (Section 1.7)

---

1. What factor has precipitated more sophisticated and complex computer-based systems?

- a. Vast increases in computer memory and storage capacity.
- b. Greater variety of exotic input/output options.
- c. Profound changes in computer architectures.
- d. All of the above.

---

8. Modern software applications are so complex that it is hard to develop mutually exclusive category names.

- a. True
- b. False

---

9. The current software crisis was caused by the Y2K problem whose seeds were first sown by careless programmers in the early 1970's.

- a. True
- b. False

---

10. Software developers succeed more often than they fail, but software failures receive more press coverage.

- a. True
- b. False

---

11. Adding more people to a project that is already behind schedule is a good way to catch up.

- a. True
- b. False

---

12. Modern CASE tools are more important than the newest hardware for achieving good software quality and productivity.

- a. True
- b. False

---

14. A general statement of objectives is all that is needed to begin developing a piece of software.

- a. True
- b. False

---

15. The formal technical review is an inadequate substitute for testing regardless of nature of the software defect.

- a. True
- b. False

16. Documentation is no longer a necessary part of the software development process because no one reads it.

- a. True
- b. False

1 CORRECT

Which of the following are recognized process flow types?

- A) Concurrent process flow
- B) Iterative process flow
- C) Linear process flow
- D) Spiral process flow
- ✓ E) both b and c

Feedback: (Section 3.1)

2 CORRECT

The communication activity is best handled for small projects using six distinct actions (inception, elicitation, elaboration, negotiation, specification, validation).

- A) True
- ✓ B) False

Feedback: (Section 3.2)

3 CORRECT

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

- A) True
- ✓ B) False

Feedback: (Section 3.3)

4 CORRECT

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

- ✓ A) True
- B) False

Feedback: (Section 3.4)

5 CORRECT

Which of these are standards for assessing software processes?

- A) SEI
- B) SPICE
- C) ISO 9000
- D) ISO 9001
- ✓ E) both b and d

Feedback: (Section 3.5)

1 CORRECT

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.

Feedback: (Section 4.1.1)

2 CORRECT

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.

Feedback: (Section 4.1.2)

3 CORRECT

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.

Feedback: (Section 4.1.3)

4 CORRECT

The prototyping model 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.

Feedback: (Section 4.1.3)

5 CORRECT

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.

Feedback: (Section 4.1.3)

6 CORRECT

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

Feedback: (Section 4.1.4)

7 CORRECT

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.

Feedback: (Section 4.2.1)

8 CORRECT

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.

Feedback: (Section 4.2.2)

9 CORRECT

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

A) Inception phase

B) Elaboration phase

C) Construction phase

✓ D) Validation phase

Feedback: (Section 4.3.2)

10 CORRECT

Which of these is not a characteristic of Personal Software Process?

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.

Feedback: (Section 4.4.1)

11 CORRECT

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

Feedback: (Section 4.4.2)

12 CORRECT

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

A) True

✓ B) False

Feedback: (Section 4.5)

13 CORRECT

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

✓ A) True

B) False

Feedback: (Section 4.6)

---

4 CORRECT

Process models are described as agile because they

A) eliminate the need for cumbersome documentation

✓ B) emphasize maneuverability and adaptability

C) do not waste development time on planning activities

D) make extensive use of prototype creation

Feedback:

---

5 CORRECT

Which of these terms are level names in the Capability Maturity Model?

A) Performed

B) Repeated

C) Reused

D) Optimized

✓ E) both a and d

Feedback:

---

1 CORRECT

The linear sequential 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 cannot be used in a modern context.

Feedback:

---

2 CORRECT

The linear sequential model of software development is also known as the

A) Classical life cycle model

B) Fountain model

C) Spiral model

D) Waterfall model

✓ E) both a and d

Feedback:

---

4 CORRECT

The rapid application development model is

A) Another name for component-based development.

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

✓ C) A high speed adaptation of the linear sequential model.

D) All of the above.

Feedback:

---

7 CORRECT

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

Feedback:

---

8 CORRECT

The concurrent development model is

A) Another name for the rapid application development model.

✓ B) Often used for the development of client/server applications.

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

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

Feedback:

---

12 CORRECT

In the Unified Process model requirements are determined iteratively and may span more than one phase of the process.

✓ A) True

B) False

Feedback:

---

2. What are the three generic phases of software engineering?

a. definition, development, support

b. what, how, where

c. programming, debugging, maintenance

d. analysis, design, testing

---

3. Which of these terms is a level name in the Capability Maturity Model?

a. Ad hoc

b. Repeatable

c. Reusable

d. Organized

4. Which of these items should be used to select a software process framework?

- a. People
- b. Product
- c. Project
- d. All of the above

---

5. In which software development problem solving stage are the results delivered?

- a. Status quo
- b. Problem definition
- c. Technical development
- d. Solution integration

---

6. Software development activities are easy to compartmentalize into four non-overlapping phases?

- a. True
- b. False

---

8. The linear sequential model of software development is also known as the

- a. Classical life cycle model
- b. Fountain model
- c. Spiral model
- d. Chaos model

---

10. The rapid application development model is

- a. Another name for component-based development.
- b. A useful approach when a customer cannot define

requirements clearly.

- c. A high speed adaptation of the linear sequential model.
- d. All of the above.

---

14. The WINWIN spiral model of software development is

- a. A used when requirements must be defined by customer negotiation.
- b. Useful when a customer is able to provide requirements completely.
- c. The best approach to use for projects with large development teams.
- d. Like the spiral model without the risk assessment step.

---

15. The concurrent development model is

- a. Another name for the rapid application development model.
- b. Often used for the development of client/server applications.
- c. Only used for development of parallel or distributed systems.
- d. Used whenever a large number of change requests are anticipated.

---

18. Fourth generation techniques

- a. Allow software to be developed without any testing.
- b. Eliminate the need for costly requirements gathering activities.
- c. Can reduce the time required to develop software.
- d. Are best used by non-programmers to build small systems.

1 INCORRECT

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

- **A)** True
- ✓ **B)** False

2 CORRECT

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

Feedback: (Section 5.1)

3 CORRECT

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

Feedback: (Section 5.2)

4 CORRECT

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

- ✓ **A)** True
- **B)** False

Feedback: (Section 5.3)

5 CORRECT

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

Feedback: (Section 5.3.2)

6 INCORRECT

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 INCORRECT

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

9 CORRECT

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

Feedback: (Section 5.5.2)

10 INCORRECT

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 CORRECT

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

5 CORRECT

Which of the following traits need to exist among the members of an agile software team?

A) Competence

B) Decision-making ability

C) Mutual trust and respect

✓ **D)** All of the above

Section 3.4.1

7
CORRECT

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

Section 3.5

9
INCORRECT

What are the three framework activities for the Adaptive Software Development (ASD) process model?

- **A)** analysis, design, coding
- **B)** feasibility study, functional model iteration, implementation
- **C)** requirements gathering, adaptive cycle planning, iterative development
- ✓ **D)** speculation, collaboration, learning

Section 3.5.2

10
INCORRECT

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 is the cause of the problems you are encountering?
- **D)** What do you plan to accomplish be the next team meeting?

Section 3.5.3

11
CORRECT

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

Section 3.5.5

12
INCORRECT

In Feature Driven Development (FDD) a client-valued feature is a client-valued function that can be delivered in two weeks or less.

- **A)** True
- ✓ **B)** False

Section 3.5.7

13
CORRECT

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

Section 3.5.8

14
CORRECT

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

1 CORRECT

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

- **A)** True
- ✓ **B)** False

Feedback: (Section 7.1)

2 CORRECT

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

Feedback: (Section 7.2)

3 CORRECT

Every communication activity should have a facilitator to make sure that the customer is not allowed to dominate the proceedings.

- **A)** True
- ✓ **B)** False

Feedback: (Section 7.3.1)

4 CORRECT

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

- ✓ **A)** True
- **B)** False

Feedback: (Section 7.3.1)

5 CORRECT

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

Feedback: (Section 7.3.2)

6 CORRECT

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

- **A)** True
- ✓ **B)** False

Feedback: (Section 7.3.2)

7 CORRECT

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

Feedback: (Section 7.3.3)

8

CORRECT

The design model should be traceable to the requirements model?

✓ A) True

B) False

Feedback: (Section 7.3.3)

9 CORRECT

Teams using agile software practices do not generally create models.

A) True

✓ B) False

Feedback: (Section 7.3.3)

10 CORRECT

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

Feedback: (Section 7.3.4)

11 CORRECT

A successful test I ones that discovers at least one as-yet undiscovered error.

✓ A) True

B) False

Feedback: (Section 7.3.4)

12 CORRECT

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

Feedback: (Section 7.3.5)

13 CORRECT

Larger programming teams are always more productive than smaller teams.

A) True

✓ B) False

Feedback: (Section 7.4)

---

5 CORRECT

Software engineers collaborate with customers to define which of the following?

A) Customer visible usage scenarios

B) Important software features

C) System inputs and outputs

✓ D) All of the above

7 INCORRECT

What role(s) do user stories play in agile planning?

A) Define useful software features and functions delivered to end-users
B) Determine a schedule used to deliver each software increment
C) Provide a substitute to performing detailed scheduling of activities
✓ D) Used to estimate the effort required build the current increment
E) both a and d

8 CORRECT

Which of the following activities is not one of the four things that need to be accomplished by the generic planning task set?

A) Develop overall project strategy
B) Identify the functionality to deliver in each software increment
✓ C) Create a detailed schedule for the complete software project
D) Devise a means of tracking progress on a regular basis

10 CORRECT

The customer can directly observe both the difference between the internal quality of a design and its external quality?

A) True
✓ B) False

12 INCORRECT

Many of the tasks from the generic task sets for analysis modeling and design can be conducted in parallel with one another.

✓ A) True
B) False

14 CORRECT

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

✓ A) True
B) False

15 CORRECT

Which of the following are tasks in the generic task set for construction?

A) Build a software component
B) Create a user interface
C) Unit test the component
D) Assess the quality of the component
✓ E) both a and c

16 CORRECT

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

1 CORRECT

Requirements engineering is a generic process that does not vary from one software project to another.

✓ A) True

B) False

Feedback: (Section 8.1)

2 CORRECT

During project inception the intent of the of the tasks are to determine

A) basic problem understanding

B) nature of the solution needed

C) people who want a solution

D) none of the above

✓ E) a, b, c

Feedback: (Section 8.1)

3 CORRECT

Three things that make requirements elicitation difficult are problems of

A) budgeting

B) scope

C) understanding

D) volatility

✓ E) b, c, d

Feedback: (Section 8.1)

4 CORRECT

A stakeholder is anyone who will purchase the completed software system under development.

A) True

✓ B) False

Feedback: (Section 8.2.1)

5 CORRECT

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

Feedback: (Section 8.2.2)

6 CORRECT

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?

Feedback: (Section 8.2.4)

7 CORRECT

Non-functional requirements can be safely ignored in modern software development projects.

A) True

✓ B) False

Feedback: (Section 8.2.5)

8 CORRECT

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

Feedback: (Section 8.3.1)

9 CORRECT

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

Feedback: (Section 8.3.2)

10 CORRECT

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

Feedback: (Section 8.3.4)

11 INCORRECT

User stories are complete descriptions the user needs and include the non-functional requirements for a software increment.

✓ A) True

B) False

12 CORRECT

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

Feedback: (Section 8.4)

13 CORRECT

Use-case actors are always people, never system devices.

A) True

✓ B) False

Feedback: (Section 8.4)

14 CORRECT

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

Feedback: (Section 8.5.1)

15 CORRECT

Analysis patterns facilitate the transformation of the analysis model into a design model by suggesting reliable solutions to common problems.

✓ A) True

B) False

Feedback: (Section 8.5.2)

16 CORRECT

In agile process models requirements engineering and design activities are interleaved.

✓ A) True

B) False

Feedback: (Section 8.5.3)

17 CORRECT

In win-win negotiation, the customer's needs are met even though the developer's need may not be.

A) True

✓ B) False

Feedback: (Section 8.6)

18 CORRECT

In requirements validation the requirements model is reviewed to ensure its technical feasibility.

A) True

✓ B) False

Feedback: (Section 8.8)

19 CORRECT

The most common reason for software project failure is lack of functionality.

A) True

✓ B) False

Feedback: (Section 8.9)

1 CORRECT

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

Feedback: (Section 9.1)

2 CORRECT

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

Feedback: (Section 9.1.1)

3 CORRECT

Object-oriented domain analysis is concerned with the identification and specification of reusable capabilities within an application domain.

✓ A) True

B) False

Feedback: (Section 9.1.3)

4 CORRECT

In structured analysis models focus on the structure of the classes defined for a system along with their interactions.

A) True

✓ B) False

Feedback: (Section 9.1.4)

5 CORRECT

Creation and refinement of use cases if an important part of scenario-based modeling.

✓ A) True

B) False

Feedback: (Section 9.2)

6 CORRECT

It is important to consider alternative actor interactions when creating a preliminary use case.

A) True

✓ B) False

Feedback: (Section 9.2.1)

7 CORRECT

Brainstorming is one technique that may be used to derive a complete set of use case exceptions.

✓ A) True

B) False

Feedback: (Section 9.2.2)

8 CORRECT

In many cases there is no need to create a graphical representation of a usage scenario.

✓ A) True

B) False

Feedback: (Section 9.2.3)

9 CORRECT

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

Feedback: (Section 9.3.1)

10 CORRECT

UML swimnlane diagrams allow you to represent the flow of activities by showing the actors having responsibility for creating each data element.

A) True

✓ B) False

Feedback: (Section 9.3.2)

---

1 CORRECT

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

Section 6.1.3

3 CORRECT

Object-oriented domain analysis is concerned with the identification and specification of reusable capabilities within an application domain.

✓ A) True

B) False

Section 6.1.4

4 CORRECT

In structured analysis models focus on the structure of the classes defined for a system along with their interactions.

A) True

✓ ○ **B)** False

Section 6.2

5 CORRECT

Creation and refinement of use cases if an important part of scenario-based modeling.

✓ ○ **A)** True

○ **B)** False

Section 6.2.1

6 CORRECT

It is important to consider alternative actor interactions when creating a preliminary use case.

○ **A)** True

✓ ○ **B)** False

---

10 CORRECT

One or more attributes of a data object must be defined as a key to allow the location of an instance of the data object.

✓ ○ **A)** True

○ **B)** False

Section 6.4.3

11 CORRECT

The entity relationship 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

Section 6.5.1

12 CORRECT

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

Section 6.5.2

13 INCORRECT

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

- **A)** True
- ✓ **B)** False

Section 6.5.3

14 CORRECT

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

Section 6.5.4

15 CORRECT

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

- **A)** class collaborators
- **B)** class name
- ✓ **C)** class reliability
- **D)** class responsibilities

Section 6.5.4

16 CORRECT

Class responsibilities are defined by

- **A)** its attributes only
- **B)** its collaborators
- **C)** its operations only
- ✓ **D)** both its attributes and operations

Section 6.5.6

17 CORRECT

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

✓ **A)** True

B) False

Section 17.1.1

1 CORRECT

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

- **A)** True
- ✓ **B)** False

Section 17.1.2

2 CORRECT

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

---

Section 17.1.3

3 CORRECT

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

- **A)** integration testing
- **B)** system testing
- **C)** unit testing
- **D)** validation testing
- ✓ **E)** c, a, d, and b

Section 17.1.4

4 INCORRECT

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

Section 17.2

5 CORRECT

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) both a and b

Section 17.3.1

6 CORRECT

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

A) algorithmic performance

B) code stability

C) error handling

D) execution paths

✓ E) both c and d

Section 17.3.1

7 CORRECT

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

A) True

✓ B) False

Section 17.3.2

8 CORRECT

Top-down integration testing has as it's 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) both b and c

Section 17.3.2

9 CORRECT

Bottom-up integration testing has as it's 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

Section 17.3.2

10 CORRECT

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)** both a and b

Section 17.3.2

11 CORRECT

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

Section 17.4.1

12 CORRECT

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

Section 17.4.2

13 CORRECT

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

Section 17.5

14 CORRECT

Since many WebApps evolve continuously, the testing process must be ongoing

as well.

✓ A) True

B) False

Section 17.6

15 CORRECT

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

✓ A) True

B) False

Section 17.6.1

16 CORRECT

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

Section 17.6.2

17 CORRECT

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

A) True

✓ B) False

Section 17.6.3

18 CORRECT

Acceptance tests are normally conducted by the

A) developer

✓ B) end users

C) test team

D) systems engineers

Section 17.7.1

19 INCORRECT

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

Section 17.7.2

20 CORRECT

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

✓ ○ **A)** True

○ **B)** False

Section 17.7.3

21 CORRECT

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

○ **A)** True

✓ ○ **B)** False

Section 17.7.4

22 CORRECT

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

○ **A)** True

✓ ○ **B)** False

Section 13.7.1

23 CORRECT

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

✓ ○ **A)** True

○ **B)** False

Section 17.8.3

24 CORRECT

Which of the following is an approach to debugging?

○ **A)** backtracking

○ **B)** brute force

○ **C)** cause elimination

○ **D)** code restructuring

✓ ○ **E)** a, b, and c

---

3 CORRECT

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

1 CORRECT

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

- **A)** True
- ✓ **B)** False

Section 18.1

2 CORRECT

Which of the following are characteristics of testable software?

- **A)** observability
- **B)** simplicity
- **C)** stability
- ✓ **D)** all of the above

Section 18.2

3 CORRECT

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

Section 18.2

4 CORRECT

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

Section 18.3

5

CORRECT

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)** both b and d

Section 18.4.1

6 CORRECT

Program flow graphs are identical to program flowcharts.

- **A)** True
- ✓ **B)** False

Section 18.4.2

7 CORRECT

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

Section 18.4.3

8 CORRECT

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

Section 18.5.1

9 CORRECT

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

Section 18.5.2

10 CORRECT

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

Section 18.5.3

11 CORRECT

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

Section 18.6

12 CORRECT

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, and c

Section 18.6.1

13 CORRECT

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

A) True

✓ ○ **B)** False

Section 18.6.2

14 CORRECT

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

Section 18.6.3

15 CORRECT

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

○ **A)** True

✓ ○ **B)** False

Section 18.6.4

16 CORRECT

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

Section 18.7

17 INCORRECT

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

Section 18.8.2

18 CORRECT

Client/server architectures cannot be properly tested because network load is highly variable.

A) True

✓ B) False

Section 18.8.4

19 CORRECT

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

A) performance

B) reliability

C) security

✓ D) time

---

14 CORRECT

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

Feedback:

16 CORRECT

Comparison testing is typically done to test two competing products as part of customer market analysis prior to product release.

A) True

✓ B) False

Feedback:

17 CORRECT

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

Feedback:

18 CORRECT

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

✓ A) True

B) False

Feedback:

19 CORRECT

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

A) True

✓ B) False

Feedback:

20 CORRECT

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

✓ A) True

B) False

Feedback:

21 INCORRECT

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

Feedback:

22 CORRECT

Testing OO class operations is made more difficult by

A) encapsulation

B) inheritance

C) polymorphism

✓ D) both b and c

Feedback:

23 INCORRECT

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

Feedback:

24 CORRECT

Deep structure testing is not designed to

A) examine object behaviors

B) exercise communication mechanisms

C) exercise object dependencies

✓ D) exercise structure observable by the user

Feedback:

25 **CORRECT**

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

✓ **A)** True

B) False

Feedback:

---

26 **INCORRECT**

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

Feedback:

---

27 **INCORRECT**

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

A) True

✓ **B)** False

Feedback:

---

28 **CORRECT**

Tests derived from behavioral class models should be based on the

A) data flow diagram

B) object-relation diagram

✓ **C)** state diagram

D) use-case diagram

Feedback:

---

29 **CORRECT**

Client/server architectures cannot be properly tested because network load is highly variable.

A) True

✓ **B)** False

Feedback:

---

18 **CORRECT**

Documentation does not need to be tested.

A) True

✓ **B)** False

Feedback: (Section 23.8)

Section 24.1

1 CORRECT

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

Section 24.1.1

2 CORRECT

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

- ✓ **A)** True
- **B)** False

Section 24.1.2

3 CORRECT

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.

Section 24.1.3

4 CORRECT

Process framework activities are populated with

- **A)** milestones
- **B)** work products
- **C)** QA points
- ✓ **D)** all of the above

Section 24.1.4

5 CORRECT

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

A) True

✓ B) False

Section 24.2.1

6 CORRECT

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

A) customers

B) end-users

C) project managers

✓ D) sales people

Section 24.2.2

7 CORRECT

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

A) True

✓ B) False

Section 24.2.3

8 CORRECT

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

Section 24.2.3

9 CORRECT

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, and c

Section 24.2.3

10 CORRECT

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.

Section 24.2.4

11 INCORRECT

Small agile teams have no place in modern software development.

○ **A)** True

✓ ○ **B)** False

Section 24.2.5

12 INCORRECT

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

○ **A)** interoperability

✓ ○ **B)** performance

○ **C)** scale

○ **D)** uncertainty

Section 24.3.1

13 INCORRECT

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

---

Section 24.3.2

14 CORRECT

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)** both b and c

Section 24.4.1

15 CORRECT

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

- ✓ **A)** True
- **B)** False

Section 24.4.2

16 CORRECT

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

Section 24.5

17 INCORRECT

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)** both c and d

Section 24.6

18 CORRECT

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, and d

Section 24.7

19 CORRECT

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, and d

Section 27.1

1 CORRECT

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

- **A)** True
- ✓ **B)** False

Section 27.1

2 CORRECT

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

Section 27.2.1

3 CORRECT

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

Section 27.2.2

4 CORRECT

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

- **A)** True
- ✓ **B)** False

Section 27.2.2

5 CORRECT

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

Section 27.2.3

6 CORRECT

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

Section 27.3

7 INCORRECT

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

Section 27.4

8 CORRECT

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) both b and c

Section 27.4

9 CORRECT

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

Section 27.5

10 CORRECT

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

○ **A)** CPM

○ **B)** DRE

○ **C)** FP

○ **D)** PERT

✓ ○ **E)** both a and d

Section 27.5.1

11 CORRECT

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

✓ ○ **A)** True

○ **B)** False

Section 27.5.2

12 INCORRECT

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

---

Section 27.5.3

13 CORRECT

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

Section 27.5.4

14 CORRECT

WebApp projects only require the creation of a macro schedule.

- **A)** True
- ✓ **B)** False

Section 27.6

15 CORRECT

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

---

Section 27.6

16 CORRECT

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

- ✓ **A)** True
- **B)** False

---

8 CORRECT

For purposes of determining the major engineering tasks and distributing them on the project time line, the project manager should assume that the process model used is

- **A)** linear
- **B)** sequential
- **C)** iterative evolutionary
- ✓ **D)** any of the above

Feedback:

9 INCORRECT

The only means accomplishing task refinement is to make use of a process design language approach.

- **A)** True
- ✓ **B)** False

Feedback: