



史上最全四川大学软件工程导论选择题题库

史上最全四川大学软件工程 导论选择题题库

-标准化文件发布号：（9456-EUATWK-MWUB-WUNN-INNUL-DDQTY-KII

Chapter 1

1

Which question no longer concerns the modern software engineer

- A) Why does computer hardware cost so much
- B) Why does software take a long time to finish
- C) Why does it cost so much to develop a piece of software
- D) Why can't software errors be removed from products prior to delivery

Section

2

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

- A) True
- B) False

Section

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

Section

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

Section

5

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

- A) Process
- B) Manufacturing
- C) Methods

D) Tools

Section

6

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

A) True

B) False

Section

7

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

Section

8

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

A) True

B) False

Section

9

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

Section

10

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

A) True

B) False

Section

11

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

- A) True
- B) False

Section

12

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

- A) True
- B) False

Chapter 2

1

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 a and c

Section

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

- A) True
- B) False

Section

3

Which of these are standards for assessing software processes

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

Section

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.

Section

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.

Section

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

Section

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.

Section

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

Section

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

Section

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.

Section

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

Section

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

Section

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

Section

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

Section

15

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

- A) True
- B)** False

Section

16

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

- A)** True
- B) False

Chapter 3.

1

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

- A) True
- B) False**

Section

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

Section

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**

Section

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

Section

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

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

Section

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

8

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

Section

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

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 by the next team meeting

Section

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

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

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

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 4

1

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

- A) True
- B) False

Section

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

Section

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

- A) True
- B) False

Section

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

- A) True
- B) False

Section

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

Section

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

- A) True
- B) False

Section

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

Section

The design model should be traceable to the requirements model

- A)** True
- B) False

Section

Teams using agile software practices do not generally create models.

- A) True
- B)** False

Section

Which of the following is not one of the principles of good coding

- A) Create unit tests before you begin coding
- B) Create a visual layout that aids understanding
- C) Refactor the code after you complete the first coding pass
- D)** Write self-documenting code, not program documentation

Section

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

- A)** True
- B) False

Section

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

Chapter 5

1

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

- A)** True
- B) False

Section

2

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

- A) basic problem understanding
- B) nature of the solution needed
- C) people who want a solution
- D) none of the above
- E) a, b, and c

Section

3

Three things that make requirements elicitation difficult are problems of

- A) budgeting
- B) scope
- C) understanding
- D) volatility
- E) b, c, and d

Section

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

- A) True
- B) False

Section

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

Section

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

Section

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

Section

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

Section

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

Section

10

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

Section

11

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

- A) True
- B) False**

Section

12

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**

Section

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

- A) True**
- B) False

Section

14

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

- A) True
- B) False**

Section

15

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

- A) True
- B) False**

Chapter 6

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

Section

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

Section

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

- A) True**
- B) False

Section

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

- A) True
- B) False**

Section

5

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

- A) True
- B) False

Section

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

- A) True
- B) False

Section

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

- A) True
- B) False

Section

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

- A) True
- B) False

Section

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

Section

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

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

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

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

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

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

Class responsibilities are defined by

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

Section

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

- A) True**
- B) False

Chapter 7

1

The data flow 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
- E) both b and c**

Section

Control flow diagrams are

- A) needed to model event driven systems.**
- B) required for all systems.
- C) used in place of data flow diagrams.
- D) used to represent system behavior.

Section

The control specification represents the system behavior using UML sequence and state diagrams.

- A) True**
- B) False

Section

The data flow diagram must be augmented by min-spec that can serve as a guide the design of the software component that will implement the process.

- A) True
- B) False

Section

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.

Section

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.

Section

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

Section

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

- A) True
- B) False

Section

9

Analysis patterns are discovered, they are not explicitly created.

- A) True
- B) False

Section

10

It is not possible to justify the time required for WebApp requirements analysis.

- A) True
- B) False

Section

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**

Section

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

- A) True**
- B) False

Section

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

Section

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

Section

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

- A) True
- B) False**

Chapter 8

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, and c**

Section

2

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

- A) accuracy
- B) complexity

- C) efficiency
- D) quality**

Section

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) both b and d**

Section

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

Section

What types of abstraction are used in software design

- A) control
- B) data
- C) environmental
- D) procedural
- E) a, b, and d**

Section

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**

Section

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

- A) True
- B) False**

Section

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

- A) True
- B) False**

Section

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

- A) True**
- B) False

Section

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.

Section

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.

Section

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

- A) True
- B)** False

Section

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

Section

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

Section

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

Section

Which design is equivalent to the floor plan of a house

- A)** Architectural design
- B) Component-level design
- C) Data design
- D) Interface design

Section

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

Section

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

Section

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

- A) True
- B)** False

Chapter 9

1

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

- A) True
- B)** False

Section

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

- A)** True
- B) False

Section

An architectural description is often documented using an architecture template.

- A) True
- B)** False

Section

4

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

- A)** True
- B) False

Section

5

An architectural style encompasses which of the following elements

- A) constraints

- B) set of components
- C) semantic models
- D) syntactic models
- E) a, b, and c**

Section

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

Section

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**

Section

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) both b and c**

Section

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, and c**

Section

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

- A) True**
- B) False

Section

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

Section

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, and c**

Section

Quantitative methods for assessing the quality of proposed architectural designs are readily available.

- A) True
- B) False**

Section

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) both b and c**

Section

15

When the overall flow in a segment of a data flow diagram is largely sequential and follows straight-line paths _____ is present.

- A) low coupling
- B) good modularity
- C) transaction flow
- D) transform flow**

Section

When a single item that triggers other data flow along one of many paths _____ characterizes the information flow in a segment of a data flow diagram is present.

- A) high coupling
- B) poor modularity
- C) transaction flow**
- D) transform flow

Chapter 10

1

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

- A) True**

- B) False

Section

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

Section

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

Section

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

- A) True
- B) False

Section

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

Section

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

- A) True
- B) False

Section

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

- A) True
- B) False

Section

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

- A) True
- B) False

Section

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, and d

Section

10

In component-level design persistent data sources refer to

- A) Component libraries
- B) Databases
- C) Files
- D) All of the above
- E) both b and c

Section

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

- A) True
- B) False

Section

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

- A) True
- B) False

Section

13

Which of these constructs is used in structured programming

- A) branching
- B) condition
- C) repetition
- D) sequence
- E) b, c, and d

Section

Which of these is a graphical notation for depicting procedural detail

- A) box diagram
- B) decision table
- C) ER diagram
- D) flowchart

Section

A decision table should be used

- A) to document all conditional statements
- B) to guide the development of the project management plan
- C) only when building an expert system

- D) when a complex set of conditions and actions appears in a component

Section

A program design language (PDL) is often a

- A) combination of programming constructs and narrative text
- B) legitimate programming language in its own right
- C) machine readable software development language
- D) useful way to represent software architecture

Section

17

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

Section

Which of the following is not one of the major activities of domain engineering

- A) analysis
- B) construction
- C) dissemination
- D) validation

Section

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

Section

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

Section

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

Section

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 11

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

Section

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, and c

Section

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

Section

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) both b and d

Section

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

- A) True
- B) False

Section

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

Section

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

Section

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

Section

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

Section

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) both c and d

Section

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

Section

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

- ☐ A) True
- ☒ B) False

Section

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

Section

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

- ☒ A) True
- ☐ B) False

Section

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

- ☒ A) True
- ☐ B) False

Section

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) both b and d

Section

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

Section

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) both b and d

Chapter 12

1

Which of the following is not one of the elements of a design pattern

- ☐ A) context
- ☐ B) environment
- ☐ C) problem

D) solution

Section

RubberNecking is an example of a classic generative pattern.

A) True

B) False

Section

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

A) True

B) False

Section

Finding patterns built by others that address design problems is often more difficult than recognizing patterns in the application to be built.

A) True

B) False

Section

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

Section

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

Section

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

A) True

B) False

Section

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

A) True

B) False

Section

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) both c and d

Section

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

- ☒ A) True
- ☐ B) False

Section

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

Section

12

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

- ☐ A) True
- ☒ B) False

Section

13

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

- ☐ A) 5
- ☒ B) 10
- ☐ C) 25
- ☐ D) 100

Section

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) both b and c

Section

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) both b and d

Section

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, and c

Chapter 13

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

Section

2

Which of the following are design goals for every WebApp

- A) Simplicity
- B) Consistency
- C) Navigability
- D) Visual appeal
- E) a, b, c, and d

Section

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

Section

4

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

- A) True
- B) False**

Section

5

Which of these are WebApp interaction mechanisms

- A) Graphic icons
- B) Graphic images
- C) Navigation menus
- D) All of the above**

Section

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

- A) True
- B) False**

Section

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

- A) True**
- B) False

Section

8

Content design is conducted by

- A) Copywriters and graphic designer
- B) Web engineers
- C) both a. and b.**
- D) none of the above

Section

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

- A) True**
- B) False

Section

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

- A) True
- B) False**

Section

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

- A) True
- B) False**

Section

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

- A) linear
- B) grid
- C) hierarchical
- D) parallel**

Section

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

Section

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

- A) True**
- B) False

Section

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**

Section

16

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

- A) True**
- B) False

Section

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

Section

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

- A) True
- B) False**

Chapter 14

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

Section

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

Section

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

- A) True
- B) False**

Section

Many software metrics can only be measures indirectly.

- A) True**
- B) False

Section

Which of the following are ISO 9126 software quality factors

- A) Functionality
- B) Portability
- C) Reliability
- D) Visual appeal
- E) a, b, and c**

Section

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

- A) True**
- B) False

Section

Software metrics represent direct measures of some manifestation of quality.

- A) True
- B) False**

Section

8

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

- A) True
- B) False**

Section

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

- A) True**
- B) False

Section

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

Section

Poor quality leads to software risks that can become serious

- A) True**
- B) False

Section

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

- A) True
- B) False**

Section

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

- A) True**
- B) False

Section

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**

Section

The project plan should include explicit techniques for _____ and _____ management

- A) change
- B) cost
- C) error

- D) quality
- E) both a and d

Section

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

Section

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

- A) True
- B) False

Chapter 15

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

Section

2

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

- A) True
- B) False

Section

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

Section

4

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

- A) True

- B) False

Section

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

- A) True
B) False

Section

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

- A) True
B) False

Section

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 and d

Section

8

An informal review may consist of which of the following

- A) casual meeting
B) desk check
C) inspection
D) pair programming
E) both a and b

Section

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

Section

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)** both a and d

Section

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

Section

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

- A)** True
- B) False

Section

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

- A) True
- B)** False

Chapter 16

1

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

- A) True
- B)** False

Section

2

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

- A)** True
- B) False

Section

3

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

- A) True
- B)** False

Section

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

Section

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

- A)** True
- B) False

Section

6

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

- A) True
- B)** False

Section

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

Section

Six Sigma methodology defines three core steps.

- A) analyze, improve, control
- B) analyze, design, verify
- C)** define, measure, analyze
- D) define, measure, control

Section

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)** both b and c

Section

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

Section

11

The ISO quality assurance standard that applies to software engineering is

- A) ISO 9000
- B) ISO 9001**
- C) ISO 9002
- D) ISO 9003

Section

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 17

1

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

- A) True
- B) False**

Section

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

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

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

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) both a and b

Section

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

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

- A) True
- B) False

Section

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

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

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

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

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

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

14

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

- A) True
- B) False

Section

15

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

Section

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

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

- A) True

B) False

Section

Acceptance tests are normally conducted by the

- A) developer
- B)** end users
- C) test team
- D) systems engineers

Section

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

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

- A)** True
- B) False

Section

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

- A) True
- B)** False

Section

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

- A) True
- B)** False

Section

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

- A)** True
- B) False

Section

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

Chapter 18

1

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

- A) True
- B) False**

Section

2

Which of the following are characteristics of testable software

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

Section

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

Section

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**

Section

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) both b and d**

Section

Program flow graphs are identical to program flowcharts.

- A) True

- B)** False

Section

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

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

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

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

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

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, and c

Section

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

- A) True
- B) False**

Section

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

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

- A) True
- B) False**

Section

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

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

Section

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

- A) True
- B) False**

Section

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

- A) performance
- B) reliability
- C) security
- D) time**

Chapter 19

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

Section

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

- A) True
- B) False

Section

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

- A) True
- B) False

Section

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

- A) True
- B) False

Section

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, and d

Section

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

- A) True
- B) False

Section

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

- A) True
- B) False

Section

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

- A) True
- B) False

Section

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

Section

Testing OO class operations is made more difficult by

- A) encapsulation
- B) inheritance
- C) polymorphism
- D) both b and c**

Section

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

Section

Deep structure testing is not design to

- A) object behaviors
- B) communication mechanisms
- C) exercise object dependencies
- D) exercise structure observable by the user**

Section

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

- A) True**
- B) False

Section

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

Section

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

- A) True
- B) False**

Chapter 20

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

Section

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

- A) True
- B) False

Section

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

Section

Test planning is not used in WebApp testing.

- A) True
- B) False

Section

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

Section

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

Section

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

- A) True
- B) False

Section

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

Section

Which of the following is not a WebApp interface mechanism

- A) Browser
- B) Cookies
- C) Forms
- D) Links

Section

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

- ☒ A) True
- ☐ B) False

Section

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

- ☐ A) True
- ☒ B) False

Section

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

Section

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

Section

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

- ☐ A) True
- ☒ B) False

Section

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

- ☒ A) True
- ☐ B) False

Section

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

Section

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

Section



版权说明：本文档由用户提供并上传，收益归属内容提供方，若内容存在侵权，请进行举报或认领

相关推荐

- 史上最全四川大学软件工

猜你想看

- 史上最全四川大学软件工
- 史上最全四川大学软件工

相关好店

fkoeqs111

「其它」

风住尘散了

「教育」

名山大川天下美食

「其它」

小风雨文档

「教育」

超越高度

「教育」

好大一棵树和远方

「教育」

工具

收藏



领福利

下载文档