一

Which question no longer concerns the modern software engineering

Why does computer hardware cost so much?

Why does software take a long time to finish?

Why does it cost so much to develop a piece of software?

Why can't software errors be removed from products prior to delivery?

..

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

True

False

..

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

True

False

..

Software deteriorates rather than wears out because

Software suffers from exposure to hostile environments.

Defects are more likely to arise after software has been used often.

Multiple change requests introduce errors in component interactions.

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

True

False

..

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

True

False

..

二

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

..

a. Process b. Manufacturing

c. Methods d. Tools

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

True

False

3. Which of these are the 5 generic software engineering framework activities? ..

communication, planning, modeling, construction, deployment

communication, risk management, measurement, production, reviewing

analysis, designing, programming, debugging, maintenance

analysis, planning, designing, programming, testing

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

a. Performed b. Repeatable

c. Reused d. Optimized

e. Both b and d

5. Which of the items listed below is not one of the process patterns. ..

a. Intent

b. Solution

c. Resulting Context

d. Output

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

a. True

b. False

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

True

False

8. The tasks (and degree of rigor) for each activity are always unchanged. ..

True

False

三

The linear sequential model of software development is

A reasonable approach when requirements are well defined.

A good approach when a working program is required quickly.

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

An old fashioned model that cannot be used in a modern context.

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

Classical life cycle model

Fountain model

Spiral model

Waterfall model

both a and d

The incremental model of software development is

A reasonable approach when requirements are well defined.

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

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

A revolutionary model that is not used for commercial products.

The rapid application development model is

Another name for component-based development.

A useful approach when a customer cannot define requirements clearly.

A high speed adaptation of the linear sequential model.

All of the above.

5. Evolutionary software process models

Are iterative in nature

Can easily accommodate product requirements changes

Do not generally produce throwaway systems

All of the above

6. 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.

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

Inception phase

Elaboration phase

Construction phase

Validation phase

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

True

False

四

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

True

False

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

Eliminate the use of project planning and testing

Only essential work products are produced

Process allows team to streamline tasks

Uses incremental product delivery strategy

How do you create agile processes to manage unpredictability?

Requirements gathering must be conducted very carefully

Risk analysis must be conducted before planning takes place

Software increments must be delivered in short time periods

Software processes must adapt to changes incrementally

Both c and d

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

Competence

Decision-making ability

Mutual trust and respect

All of the above.

5. 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

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

a. analysis, design, coding, testing

b. planning, analysis, design, coding

c. planning, analysis, coding, testing

d. planning, design, coding, testing

7. 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

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

a. What did you do since the last meeting?

b. What obstacles are you encountering?

c. What is the cause of the problems you are encountering?

d. What do you plan to accomplish at the next team meeting?

五

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

True

False

..

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

True

False

..

Which of the following is not one of the context-free questions that would be used during project inception?

What will be the economic benefit from a good solution?

Who is against this project?

Who will pay for the work?

Who will use the solution?

..

The use of traceability tables helps to

debug programs following the detection of run-time errors

determine the performance of algorithm implementations

identify, control, and track requirements changes

none of the above

..

5. The system specification describes the

Function, performance and constraints of a computer-based system

implementation of each allocated system

element software architecture

time required for system simulation

..

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

a. True

b. False

..

7. Which of the following is not one of the requirement classifications used in Quality Function Deployment (QFD)?

exciting

expected

mandatory

normal

..

六

Which of these is not an element of an object-oriented analysis model?　..

Behavioral elements

Class-based elements

Data elements

Scenario-based elements

Which of the following is not an objective for building an analysis model? ..

define set of software requirements that can be validated

describe customer requirements

develop an abbreviated solution for the problem

establish basis for software design

The data flow diagram ..

depicts relationships between data objects

depicts functions that transform the data flow

indicates how data are transformed by the system

indicates system reactions to external events

both b and c

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

class collaborators

class name

class reliability

class responsibilities

5. For purposes of behavior modeling a state is any

consumer or producer of data.

data object hierarchy.

observable mode of behavior.

well defined process.

..

6. Attributes cannot be defined for a class until design has been completed.

a. True

b. False

..

7. Operations are object procedures that are invoked when an object receives a message.

True

False

..

8. 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

..

七

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

architecture

data

interface

project scope

a, b and c

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

exhibits strong coupling between its modules

implements all requirements in the analysis model

includes test cases for all components

provides a complete picture of the software

both b and d

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

True

False

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

can be written more compactly.

focuses on just one thing.

is able to complete its function in a timely manner.

is connected to other modules and the outside world.

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

can be written more compactly.

focuses on just one thing.

is able to complete its function in a timely manner.

is connected to other modules and the outside world.

6. Polymorphism reduces the effort required to extend an object system by ..

coupling objects together more tightly.

enabling a number of different operations to share the same name

making objects more dependent on one another.

removing the barriers imposed by encapsulation.

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

Architectural design elements

Component-level design elements

Data design elements

Interface design elements

8. Which design is analogous to the floor plan of a house? ..

a. Architectural design elements

b. Component-level design elements

c. Data design elements

d. Interface design elements

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

Architectural design elements

Component-level design elements

Data design elements

Interface design elements

..

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

a. Architectural design elements

b. Component-level design elements

c. Data design elements

d. Interface design elements

..

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

a. True

b. False

..

12. One of the key problems in software reuse is the inability to find existing reusable design patterns when hundreds of candidates exist.

a. True

b. False

..

八

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

constraints

set of components

semantic models

syntactic models

a, b and c

During the process of modeling the system in context, systems that interact with the target system are not represented as ..

Peer-level systems

Subordinate systems

Super-ordinate systems

Working systems

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

low coupling

Good modularity

transaction flow

transform flow

When you encounter both transform flow and transaction flow in the same DFD the flow is partitioned and the appropriate mapping technique is used on each part of the DFD. ..

True

False

5. When a single item that triggers other data flow along one of many paths of a data flow diagram, \_\_\_\_\_\_\_\_ characterizes the information flow.　　..

high coupling

poor modularity

transaction flow

transform flow

6. In transaction mapping the first level factoring results in the ..

creation of CFD.

derivation of control hierarchy

distribution of work modules

refinement of the module view

7. A successful application of transform or transaction mapping to create an architectural design is supplemented by ..

entity relationship diagram

module interface descriptions

processing narratives for each module

test case for each module

Both b and c

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

a. True

b. False

九

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

attributes and operations

instances of each class

roles for each actor (device or user)

a set of collaborating classes

..

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

Control component

Infrastructure component

Problem domain component

All of the above

..

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

Dependency Inversion Principle

Parsimonious Complexity Principle

Interface Segregation Principle

Open-Closed Principle

..

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

True

False

5. In component design, elaboration does not require which of the following elements to be described in detail?

Source code

Attributes

Interfaces

Operations

b, c and d

..

6. In component-level design "persistent data sources" refer to ..

Component libraries

Databases

Files

All of the above

Both b and c

7. The object constraint language (OCL) complements UML by allowing a software engineer to use a formal grammar to construct unambiguous statements about design model elements.

True

False

..

8. Which of these criteria are useful in assessing the effectiveness of a particular design notation?

a. maintainability b. modularity c. simplicity d. size e. a, b, and c

..

十

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

allow interaction to interruptible

allow interaction to be undoable

hide technical internals from casual users

only provide one defined method for accomplishing a task

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

define intuitive shortcuts

disclose information in a progressive fashion

establish meaningful defaults

provide an on-line tutorial

answers a, b and c

Interface consistency implies that ..

each application should have its own distinctive look and feel

input mechanisms remain the same throughout the application

navigational methods are context sensitive

visual information is organized according to a design standard

both b and d

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

True

False

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

design model

implementation model

user model

user's model

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

cost estimation

interface construction

interface validation

user and task analysis

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

have users indicate their preferences on questionnaires

rely on the judgement of experienced programmers

study existing computer-based solutions

observe users performing tasks manually

both c and d

8. 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

十一

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

integration testing, unit testing, system testing, validation testing

validation testing, unit testing, integration testing, system testing

unit testing, integration testing, validation testing, system testing

system testing, validation testing, integration testing, unit testing

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

conduct formal technical reviews prior to testing

specify requirements in a quantifiable manner

use independent test teams

wait till code is written prior to writing the test plan

answers a and b

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

algorithmic performance

code stability

error handling

execution paths

both c and d

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

True

False

5. Top-down integration testing has as it‘s major advantage(s) that　　　 ..

low level modules never need testing

major decision points are tested early

no drivers need to be written

no stubs need to be written

both b and c

6. Bottom-up integration testing has as it's major advantage(s) that ..

major decision points are tested early

no drivers need to be written

no stubs need to be written

regression testing is not required

7. The OO testing integration strategy involves testing ..

groups of classes that collaborate or communicate in some way

single operations as they are added to the evolving class implementation

operator programs derived from use-case scenarios

none of the above

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

a. backtracking

b. cause elimination

c. brute force

d. code restructuring

e. a, b and c

十二

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

observability

simplicity

stability

all of the above

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

black-box testing

glass-box testing

grey-box testing

white-box testing

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

behavioral testing

black-box testing

grey-box testing

white-box testing

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

cycles in the program

errors in the program

independent logic paths in the program

statements in the program

..

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

incorrect or missing functions

interface errors

performance errors

all of the above

none of the above

6. Testing OO class operations is made more difficult by ..

encapsulation

inheritance

polymorphism

both b and c

7. What is the differences between black-box testing and white-box testing?

..lack-box testing involves testing the functionality of a software component without knowing the details of its internal logic. White-box testing involves testing the independent logic paths with full implementation knowledge.

8. What is equivalence partitioning as it applies to software testing? What is scenario-based testing?

..quivalence partitioning technique divides the input domain into classes of equivalent data items. Test cases are derived from combinations of elements from each equivalence class. Exhaustive testing of all input domain values is not necessary. Scenario-based testing: The user tasks described in the use-cases are used to construct the test cases. It is used to uncover errors that occur when actors interact with the software (focus is on user behavior, not product behavior).