# 软工必刷选择题

Karry Ren

必须全部背诵的题目

# 1 第一章 Introduction

## 1.1 选择题

1. Which question no longer concerns the modern software engineering

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?

【Answer: a】

2. Today the increased power of the personal computer has brought about an abandonment of the practice of team development of software 【False】 团队开发永远在

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

Answer: c 【deteriotates is the mean of "退化"】

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.

Answer: d 【custom 定制的 现成的组件很可能是不能用的】

6. The nature of software applications can be characterized by their information

a. complexity

b. content

c. determinacy

d. both b and c

Answer: d 【肯定是从内容和确定性来看】

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

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

# 2 第二章 软件过程

## 2.1 基础概念

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

a. Process    b. Manufacturing

c. Methods    d. Tools

【b 朴实无华的考点】

2. Software engineering umbrella activities are only applied during the initial phases of software development projects. 【False 全过程】

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

a. communication, planning, modeling, construction, deployment
b. communication, risk management, measurement, production, reviewing
c. analysis, designing, programming, debugging, maintenance
d. analysis, planning, designing, programming, testing

【a 会背就会】

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】
【Explain】 总共有 five levels:
1. initial 2. repeatable 3. defined 4. managed 5. optimized

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

a. Intent
b. Solution
c. Resulting Context
d. Output

【d】Process patterns define a set of activities, actions, work tasks, work products and/or related behaviors

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

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

8. The tasks (and degree of rigor) for each activity are always unchanged. Answer: 【False】

## 2.2 具体模型

1. The linear sequential model （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 cannot be used in a modern context.

a. 【软件设计过程的特点】

2. 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

【e】

3. 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.

【b】

4. 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.

【c】

5. 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

【d】

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.

【b】

7. 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

【d】有以下几个阶段：

1. inception 2. elaboration 3. construction 4. transition 5. production

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

## 3 第三章：敏捷过程

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

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

```
a. Eliminate the use of project planning and testing
b. Only essential work products are produced
c. Process allows team to streamline tasks
d. Uses incremental product delivery strategy
```

【a】 不需要评估用处

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
```

【e】

4. 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.
```

【d】

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

`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
```

【d】

`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
```

【d】

`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?
```

【c】

## 4 对应课本第七章：理解需求

`1.` In requirements validation the requirements model is reviewed to ensure its technical feasibility 【False】

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

`3.` 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 against this project?

c. Who will pay for the work?

d. Who will use the solution?

【b】

4. The use of traceability tables helps to

a. debug programs following the detection of runtime errors

b. determine the performance of algorithm implementations

c. identify, control, and track requirements changes

d. none of the above

【c】

5. The system specification describes the

a. Function, performance and constraints of a computer-based system

b. implementation of each allocated system

c. element software architecture

d. time required for system simulation

【a】

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

7. 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

【c】

## 5 对应课本第八章 需求建模

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

a. Behavioral elements

b. Class-based elements

c. Data elements

d. Scenario-based elements

【c】

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

【c】

3. 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

【e】

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

a. class collaborators

b. class name

c. class reliability

d. class responsibilities

【c】

5. 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.

【c】

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

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

8. UML activity diagrams are useful in representing which analysis model elements?

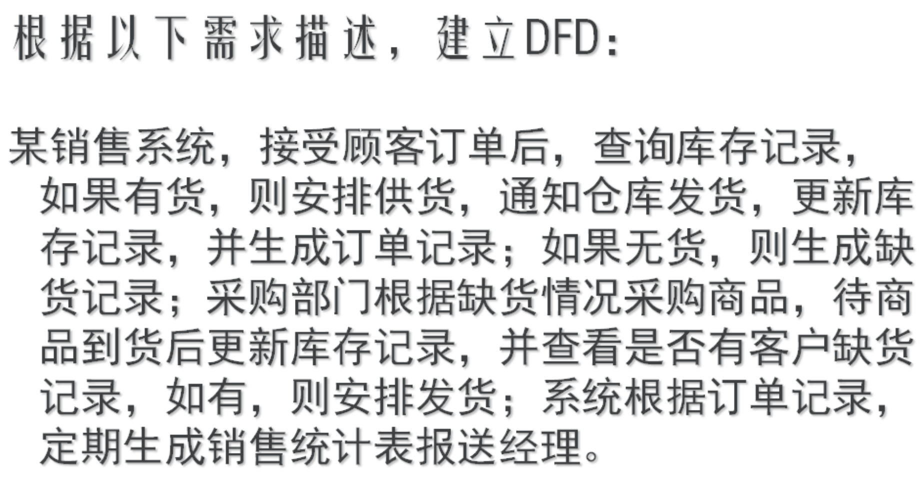
a. Behavioral elements

b. Class-based elements

c. Flow-based elements

d. Scenario-based elements

【d】



根据以下需求描述，建立DFD：

某销售系统，接受顾客订单后，查询库存记录，如果有货，则安排供货，通知仓库发货，更新库存记录，并生成订单记录；如果无货，则生成缺货记录；采购部门根据缺货情况采购商品，待商品到货后更新库存记录，并查看是否有客户缺货记录，如有，则安排发货；系统根据订单记录，定期生成销售统计表报送经理。

## 6 设计的基础概念

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

a. architecture

b. data

c. interface

d. project scope

e. a, b and c

【e】

1. component 2. architecture 3. interface 4. data

2. 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

【e】

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

4. 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.

【b】

5. 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.

【d】

6. Polymorphism reduces the effort required to extend an object system by （多态性）

a. coupling objects together more tightly.

b. enabling a number of different operations to share the same name

c. making objects more dependent on one another.

d. removing the barriers imposed by encapsulation.

【b】

7. 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

【c】

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

【a】

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

a. Architectural design elements

b. Component-level design elements

c. Data design elements

d. Interface design elements

【d】

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

【b】

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

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

## 6.1 Architecture Design

1. 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

【e】

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

a. Peer-level systems

b. Subordinate systems

c. Super-ordinate systems

d. Working systems

【d】 背下来就好了

3. 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

【d】 背下来就好了

4. 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】 背下来

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

a. high coupling
b. poor modularity
c. transaction flow
d. transform flow

【c】

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

a. creation of CFD.
b. derivation of control hierarchy
c. distribution of work modules
d. refinement of the module view

【b】

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

a. entity relationship diagram
b. module interface descriptions
c. processing narratives for each module
d. test case for each module
e. Both b and c

【e】

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

## 6.2 Component Level Design

1. 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. a set of collaborating classes
```

【d】

2. 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
```

【d】

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

```
a. Dependency Inversion Principle
b. Parsimonious Complexity Principle
c. Interface Segregation Principle
d. Open-Closed Principle
```

【b】

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

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

a. Source code

b. Attributes

c. Interfaces

d. Operations

e. b, c and d

【a】

6. 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

【e】

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】

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

a. maintainability  b. modularity  c. simplicity

【a\b\c】

## 7 用户体验设计

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 defined method for accomplishing a task

【d】

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

a. define intuitive shortcuts

b. disclose information in a progressive fashion

establish meaningful defaults

c. provide an on-line tutorial

d. answers a, b and c

【d】

3. 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

【e】

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

5. 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

【c】

6. 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

【a】

7. 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

【e】

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

【e】

## 8 软件测试

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

a.integration testing， unit testing， system testing， validation testing

b.validation testing， unit testing， integration testing， system testing

c.unit testing， integration testing， validation testing， system testing

d.system testing， validation testing， integration testing， unit testing

【c】

2. 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. answers a and b

【e】

3. 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

【e】

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

5. 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

【e】

6. 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

【c】

7. 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

【a】

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

【e】

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

a. observability

b. simplicity

c. stability

d. all of the above

【d】

10. 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

【a】

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

white

12. 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

【c】

13. 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. all of the above

e. none of the above

【d】

14. Testing OO class operations is made more difficult by

a. encapsulation

b. inheritance

c. polymorphism

d. both b and c

【d】