《软件工程概论》课后练习题

**Chapter 1 Why Software Engineering ?**

Part 1 Fill Blanks

1. Software is a set of ， and 。
2. Software engineering means the application of a ， and
3. approach to the ， ， and ，of software. That is, the application of engineering to software.
4. Any Problem-solving technique must have two parts, to problem to determine its nature, and then a solution.
5. The McCall’s quality model concerns the quality of ，CMM concerns the quality of ，and ROI concerns the quality of .
6. The is the company, organization ,or a person who is paying for the software system to be developed. The the company, organization ,or a person who is building the software system. The is the person or people who will actually use the system.
7. Any entity to be engineered ,we must do ， ， ， ，and 。

Part 2 Brief Description

Briefly describe the roles of analyst, designer, programmer, tester, and trainer.

**Chapter 2 Modeling the Process and Life Cycle**

Part 1 Fill Blanks

1. We can think a set of ordered tasks as a , a series of , including ， and 。
2. When the process involves building of some , we sometime refer to the Process as a 。
3. The life cycle of a software product include ， ， ， ，and 。
4. The software development usually involves ， ， ， ， ， ， ， and stages。
5. The waterfall model include ， ， ， ， ， ， and steps.
6. The ensures that the system has implemented all of the requirement, But the ensures that each function works correctly.
7. The V model makes more explicit some of the and that are hidden In the waterfall model。
8. The spiral model combine the development activities with management to minimize and control the 。
9. A is a partially developed product that enable customers and developers
10. To examine some aspect of the proposed system and decide if it is suitable or appropriate for the final product.

Part 2 Brief Description

Briefly describe the advantage and disadvantage of the WatterFall model.

**Chapter 3 Planning and Managing the Project**

Part 1 Fill Blanks

1. A describe the software development cycle for a particular project by enumerating the or stages of the project and breaking each into discrete or to be done.
2. The Schedule is a that shows when activities will begin and end, and When the related development products will be ready.
3. The Deliverables , that is the that the customer expects to see during project development.
4. An is a part of the project that takes place over o period of time, whereas a is the completion of an activity- a particular point of time.
5. The depicts the project as a set of discrete pieces of work.
6. The or for an activity is the estimated amount of time required for the activity to be completed. The is the amount of time available in the schedule for the activity’s completion. or for an activity is the difference between the and for that activity
7. The is a path that the slack time at every node is zero.
8. GANTT CHART is used to depict the projects’ .

Part 2 Brief Description and Exercises

1. Gave out some deliverables.
2. Briefly describe the characteristics of software development team’s individual.
3. Do the exercise 2, and 3 of this chaper ( Forth Edition PP. 138-139)

**Chapter 4 Capture the Requirements ?**

Part 1 Fill Blanks

1. A requirement is an of 。
2. The process for capturing the requirement has for steps, there are ， , ,and 。
3. The requirement captures from seven system stakeholders, they are , , , , , ,and .
4. The sources of possible requirements are the , , , , , and the .
5. There four type of requirement, they are the , , and .
6. There are two kinds of requirement documentations, they are and .
7. There are three core constructs in ERD, they are the , and .
8. There are four core construct in DFD, they are the , , and .
9. In UCD, a large box represents the system .
10. The two approaches to prototype, they are the and .

Part 2 Brief Description and Exercises

1. Briefly describe the roles of the seven groups of stakeholders.
2. Briefly describe the function of the four types of the requirement.
3. Briefly describe the functions of three core construct of ERD.
4. Briefly describe the functions of the two approaches of prototyping.
5. Do the Exercise 12 (only draw out USD) of this chapter.
6. Do the exercise 13 of this chapter.

**Chapter 5 Design the Architecture**

Part 1 Fill Blanks

1. Design is the creative process of figuring out how to implement all the customer’s requirements; the plan is called the design.
2. The software architecture is consisted by three parts, they are ,

and .

1. In pipe-and-filter style, the functions is to pass input data through a sequence of data-transforming components, and the simply transmit data from one filter to the next without modifying the data.
2. In client-server architecture, the component offer services, and access them using a request/reply protocol.
3. In peer-to-peer architecture, each executes as its own process and acts as both a client and a server to other peer component.
4. In publish-subscribe architecture, A component express interest in an event by to it. When another component that the event has take place, the subscribing component are notified.
5. A repository style consists of two types of component, one is called , another is called component.

Part 2 Brief Description and Exercise

1. Briefly describe the process for developing a software architecture.
2. Drawn out(画出) the layout architecture of ISO model for network communication.
3. Do the exercise 2 of this chapter.

**Chapter 6 Design the Modules**

Part 1 Fill Blanks

1. Design modules means decide how the will be designed

at a modular level so that developers can that the design.

2．This chapter can help you see why certain design and are

applicable, and then assist you in deciding when should apply them.

3. In practice, there is no between the end of the architectural design

Phase and the start of the module design phase.

4. Design principles are guidelines for our system’s required functionality

and behavior into modules.

5. Modularity, also called , is the principle of keeping separating

The various unrelated aspects of the system.

6. There are six types of coupling, they are , , ,

, ,and .

7. There are seven types of cohesion, they are , , ,

, , , and .

8. When complex data structures are passed between modules, we say that

There is between the modules, if only data value, and not structured

data, are passed, then the modules are connected by .

9. The modules is , in that a module’s data and functions are related

only because they are used at the same time at execution.

Part 2 Brief Description and Exercises

1. Do the exercise 7 of this chapter ( Fourth Edition P370)
2. Do the exercise 8 of this chapter ( Fourth Edition P370)

**Chapter 7 Writing the Programs**

Part 1 Fill Blanks

1. Even when writing the code itself, many people are usually involved, and a great deal of and is required.
2. May companies insist that their code conform to , , and

standards, so that the code and associated are clear to everyone

who them.

1. No matter how what language is used, each program component involves

At least three major aspects, , , and .

1. The program to be the set of written descriptions that explain to a

reader what the programs to and how they do it. is descriptive material

written within the code, all other documentation is .

1. The external documentation is the full blown report, it answers the same questions, , , , , , and using a system rather than

a component perspective.

Part 2 Brief Description and Exercise

1. Briefly describe the content of Head Block Comment.
2. Do the exercise 6 of this chapter( Forth Edition P. 399)

**Chapter 8 Testing the Programs**

Part 1 Fill Blanks

1. A fault occurs when a human makes , called an error, in performing some software behavior. Thus a fault is an view of system, as seen by the eyes of the .
2. A failure is a for the system’s required behavior. Thus a failure is an view: a problem that the sees.
3. Fault identification is the process of what the fault or faults Caused the failure, and fault correction or removal is the process of to the system so that the faults are removed.
4. An algorithm fault occurs when a component’s or does not produce the proper output for a given input.
5. Computation and precision faults occur when a formula’s Wrong or dose not compute the result to the required number of places.
6. When the documentation does not match what the program actually Dose, we say that the program has .
7. Stress or overload fault occur when the data structures are filled past their Specified .
8. A test evaluates the system to determine if the functions described by the requirements specification are actually performed by the integrated system.
9. A test compares the system with the remainder of these software and hardware requirements.
10. A test checks against the customer’s requirement description, and This test complete jointly with the customer.
11. A test is run to make sure that the system still functions as it should.
12. We view the test object form the outside as a closed or black box whose contents are unknown, our test inputs to the closed box and notes what is produced.
13. We view the test object as an open box and or white box, we can see the of the test object to test in different way.
14. A test point or test case is a particular choice of to be used in Test a program.
15. In bottom-up integration test, we should develop a to passes a test case to the component to be tested.
16. In Top-down integration test, we write a to simulate the activity of the missing component.

Part 2 Brief Description and Exercises

1. Simply describe the code walkthroughs and code inspection .
2. If we wanted to test a positive input value, given the properly test case.
3. Do the 7 of this chapter( only use bottom-up and top-down

Approaches， Fourth Edition P451).

1. Figure 1. is a program’s logic flow, give out:
2. the test case for STAEMENT TESTING, BRANCH TESTING,
3. the test path for PATH TESTING.

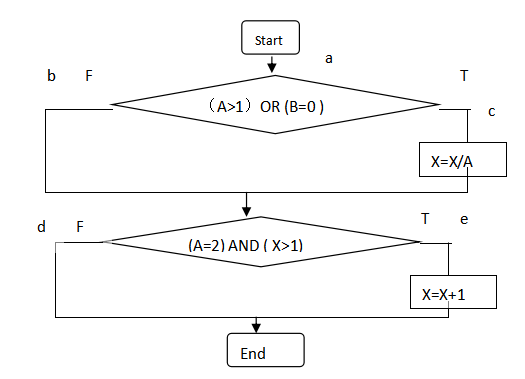


Figure 1. Program Logic Flow

**Chapter 9 Testing the System**

Part 1 Fill Blanks

1. A system configuration is a collection of system delivered

To a particular customer.

1. The software reliability is a possibility of the system will operate without

The failure under a given and a given interval.

1. The software availability is the probability that a system is operating successfully according to specification at a given of time.
2. The software maintainability is the probability that , for a given condition

Of use, a maintenance activity can be carried out within stated interval and using stated and .

1. The test in develop environment is called test, and in customer’s

environment is called test.

Part 2 Brief Description and Exercise

1.Briefly describe the content of :

1. Test Plan
2. Test Specification
3. Test Description
4. Test Report

**Chapter 10 Delivering the System**

Part 1 Fill Blanks

1. The exercises the main system functions, and the performs

The system functions to support the major work.

Part 2 Brief Description and Exercise

1. Briefly describe the functions of the user and the operator.
2. Briefly describe the training content for user training, operator training and the special training.

**Chapter 11 Maintaining the System**

Part 1 Fill Blanks

1. Any work done to the system after it is in operation is considered to be maintenance.
2. The S-systems are defined by and derivable from a specification.
3. The P-system are very abstract, and is almost impractical and

Impossible.

1. To control the day-to-day system functions, we on the maintenance team respond to problems from faults. This kind of maintenance is called

Maintenance .

1. Suppose the existing database management system is upgrade to

A new version, this maintenance is called maintenance.

1. If the customer wanted to add a new function, this kind of maintenance

Is called maintenance.

Part 2 Brief Description and Exercise

1. List the factors affecting the maintenance cost.