软件工程

作业

课　　程　：　软件工程

专　　业　：　计算机科学与技术

姓　　名　：　白文强

学　　号　：　20191060064

任课教师：　金钊

第 1 章 1.16 节练习题：6

6. Many organizations buy commercial software, thinking it is cheaper than developing and maintaining software in-house. Describe the pros and cons of using COTS software. For example, what happens if the COTS products are no longer supported by their vendors? What must the customer, user, and developer anticipate when designing a product that uses COTS software in a large system?

Pros:

A significant advantage of using COTS is that the time needed to purchase this software is much shorter than the time it takes to develop it. As such, it uses fewer resources like human capital, office space, and money. COTS also has a greater chance of incorporating industry standards. They depend less on the platform, which means excellent reliability across environments and uses.

By using COTS software, the organizations could increase reliability, reduce cost, and keep delivery times short in software systems development.

Cons:

Using COTS, though, brings significant drawbacks. For one, a vendor could cease support for it, or the software developers could go out of business, so the customer would have to look for different software.

Furthermore, the software may need customisation to fit specific business functionalities, defeating the purpose of getting something ready-made. Finally, when you have the same COTS software for several users or using one with periodic licensing, it could be more expensive in the long run.

Customer:

Customers must anticipate that the use of COTS may increase the speed of software development and reduce the cost of software. But at the same time, if COTS stops maintenance, it will make the project development cycle longer and the cost higher.

User:

Users must anticipate that the system uses COTS during the development process, then the system may not be updated for a period of time because COTS stops maintenance, and it may be difficult to use

Developer:

Developers must expect that developing a large system will require a long cycle of

using COTS. During project development, COTS may stop maintenance and become unavailable, resulting in stagnation of the development process and replacement of new COTS products, thereby increasing the difficulty of development. , Increase development time.

第 2 章 2.12 节练习题：11

11.Consider the processes introduced in this chapter. Which ones give you the most flexibility to change in reaction to changing requirements?

Phased development allows the most flexibility when changing requirements.

Incremental development are good at handling requirements changes that add or remove entire functional areas, as each increment increases the functionality of the system. Therefore, each new feature is added to the requirements, and its implementation can be planned as part of future increments. If a function is removed, it may not even be implemented and can be removed from the plan for future increments.

Iterative models are good at handling modification requirements. During iterative development, each function is implemented at the beginning, but refined through successive iterations. Since most functions are modified in each iteration, it is usually not difficult to incorporate modifications due to changing requirements.

第 3 章 3.14 节练习题：5，11

5.Describe how adding personnel to a project that is behind schedule might make the project completion date even later.

If the project adds people, then someone needs to interface with the new members so that they can know the framework of the project and the specific details. This adds extra time and increases the cost of communication. These costs may add up to more than the time saved by the addition of staff.

11. Even on your student projects, there are significant risks to your finishing your project on time. Analyze a student software development project and list the risks. What is the risk exposure? What techniques can you use to mitigate each risk?

Student software development projects are characterized by students' immature skills and little project experience, which makes them prone to inconsistencies and unrealized requirements during development. This leads to situations where the project needs to be refactored, or where there are major bugs in the software.

Risks:

- Continuing stream of requirements changes
- Unrealistic schedules and budgets
- Developing the wrong software functions

Risk	P(UO)	L(UO)	Risk Exposure
Continuing stream of requirements changes	0.8	¥2000	1600
Unrealistic schedules and budgets	0.6	¥1500	900
Developing the wrong software functions	0.2	¥1800	360

To reduce the risk, students could using incremental development to deferring changes to later increments. During the design phase, detailed, multi-source cost and schedule estimates for projects should be used in order to reduce the possibility of the unrealistic schedules and budgets. Before writing the code, a prototype should be created so that catch errors earlier.

第 4 章 4.19 节练习题：3

3.In an early meeting with your customer, the customer lists the following “requirements” for a system he wants you to build:

(a) The client daemon must be invisible to the user

(b)The system should provide automatic verification of corrupted links or outdated data

(c) An internal naming convention should ensure that records are unique

(d) Communication between the database and servers should be encrypted

(e) Relationships may exist between title groups [a type of record in the database]

(f)Files should be organizable into groups of file dependencies

(g) The system must interface with an Oracle database

(h) The system must handle 50,000 users concurrently

Classify each of the above as a functional requirement, a quality requirement, a design constraint, or a process constraint. Which of the above might be premature design decisions? Re express each of these decisions as a requirement that the design decision was meant to achieve.

(a) The client daemon must be invisible to the user.

Design constraint.

(b) The system should provide automatic verification of corrupted links or outdated d ata.

Functional requirement.

(c) An internal naming convention should ensure that records are unique.

Functional requirement.

(d) Communication between the database and servers should be encrypted.

Functional requirement.

(e) Relationships may exist between title groups [a type of record in the database].

Design constraint.

(f) Files should be organizable into groups of file dependencies.

Functional requirement

(g) The system must interface with an Oracle database.

Design constraint

(h) The system must handle 50,000 users concurrently.

Quality requirement.

(a) and (c) refer to design constructs. These requirements could be better expressed by eliminating reference to these constructs:

a) User should believe that he is interacting with a centralized system

c) All records must be unique

第 5 章 5.18 节练习题： 10，11

10. You have been hired by a consulting firm to develop an income tax calculation pa

ckage for an accounting firm. You have designed a system according to the customer’s requirements and presented your design at an architectural review. Which of the following questions might be asked at the review? Explain your answers.

(a) What computer will it run on?

(b) What will the input screens look like?

(c) What reports will be produced?

(d) How many concurrent users will there be?

(e) Will you use a multiuser operating system?

(f)What are the details of the depreciation algorithm?

Questions c,d,e may be asked. All three of these questions relate to the design of the software architecture. Question c:The type of report affects the design of the report module and how it interacts with other modules; questions d and e both involve the system's support for concurrency and should be addressed in the design of the architecture.

Questions a,b,f will not be asked. Question a: The details of the deployment platform of the system should not be considered in the design of the system architecture; Question b: There is no relationship between the design of the interaction between the system and the user and the architecture; Question f: The implementation details of specific algorithms are not a problem to be considered in the system architecture.

11. For each of the systems described below, sketch an appropriate software architecture and explain how you would assign key functionalities to the design’s components.

(a) a system of automated banking machines, acting as distributed kiosks that bank customers can use to deposit and withdraw cash from their accounts

(b) a news feeder that notifies each user of news bulletins on topics in which the user has

(c)image-processing software that allows users to apply various operations to modify their pictures (e.g.， rotation, color tinting, cropping)

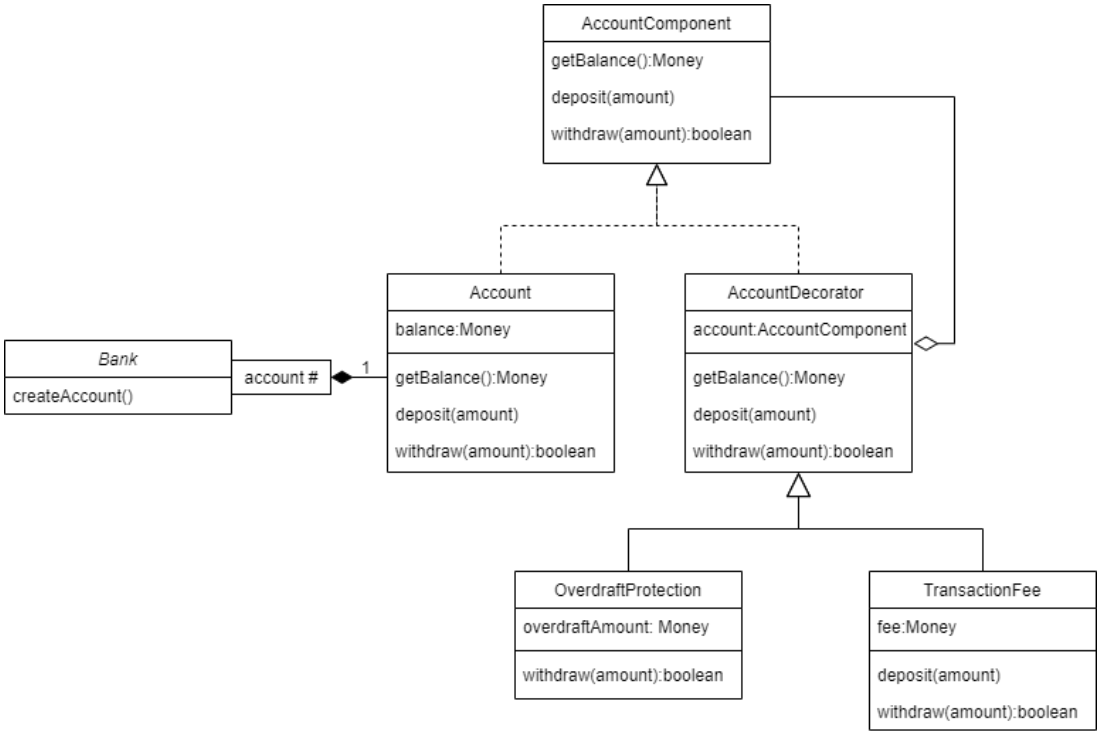
(d) a weather forecasting application that analyzes tens of thousands of data elements

collected from various sensors; the sensors periodically transmit new data values

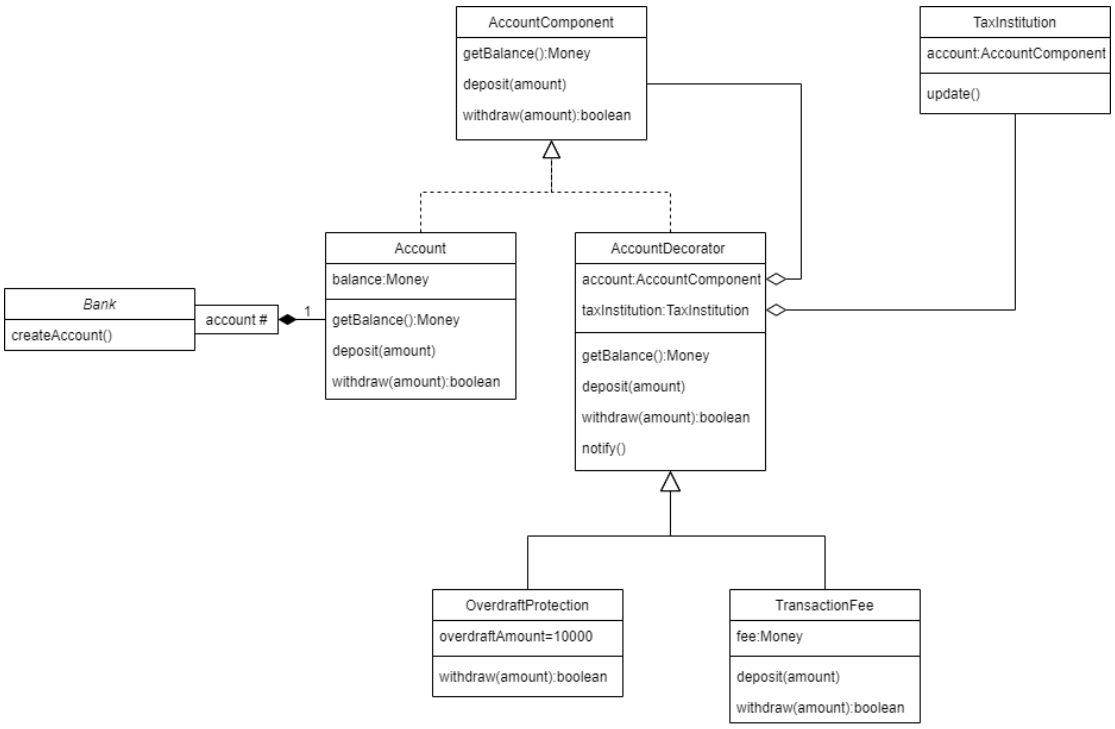
(a) **Repositories.** The central data store (bank server) contains information about all accounts in the bank. Each machine is a data accessor, capable of retrieving account information for a given account and withdrawing or depositing cash into that account.
(b) **Publish-Subscribe**. Every user can subscribe the news they needed, and when the news are published, users who subscribed the news will get a notice.
(c) **Pipe-and-Filter.** Each operation corresponds to a filter that outputs the user input image as target data. The filters could be combined to process the images.
(d) **Cilent-Server**. Every sensor is a client, they generate data and send it to the central server.

第 6 章 6.16 节练习题：22，23

22. Consider a simplified OO design, shown in Figure 6.47, for a banking system. Accounts can be created at the bank, and money can be deposited and withdrawn from the account. An account is accessed by its account number. Use the Decorator design pattern to add two new banking features to the design:
(a) Overdraft protection: allows the customer to withdraw money when the account balance is zero; the total amount that can be withdrawn in this feature is a predefined credit limit.
(b) Transaction fee: charges the customer a fixed fee for each deposit and withdrawal transaction.



23. A bank must report to the government's tax institution all transactions (deposits and withdrawals) that exceed $10,000. Building on the initial design of the banking system from question 22, use the Observer design pattern to construct a class that monitors all Account transactions.



第 8 章 8.16 节练习题：4，7

4. Suppose a program contains N decision points, cach of which has two branches. How many test cases are needed to perform path testing on such a program? If there are M choices at each decision point, how many test cases are needed for path testing? Can the program's structure reduce this number? Give an example to support your an swer.

If there are N decision points and each decision points has 2 branches, then the total number of possible paths to pass all the decision points is 2^N.So the total number of the test cases is 2^N.

If each decision points have M choices, then total number of paths to pass all decision points is M^N;

The effect of reducing the number of paths can be achieved by nested branches. For example, the following four branches have a total of 2^4=16 paths

if(a<0)and(b<0)then…
if(a<0)and(b>=0)then…
if(a>0)and(b<0)then…
if(a>0)and(b>=0)then…

The above program is equivalent to the following program, but the latter has only four paths, only four test cases are needed

```
if(a<0)then
    if((b<0) then…
    else…
else
    if((b<0) then…
    else…
```

7. Figure 8.22 illustrates the component hierarchy in a software system. Describe the s equence of tests for integrating the components using a bottom-up approach, a top-down approach, a modified top-down approach, a big bang approach, a sandwich approach, and a modified sandwich approach.

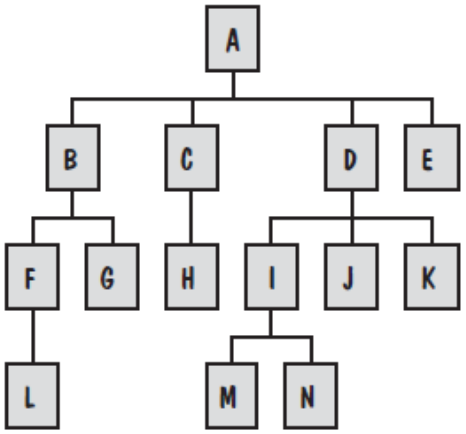
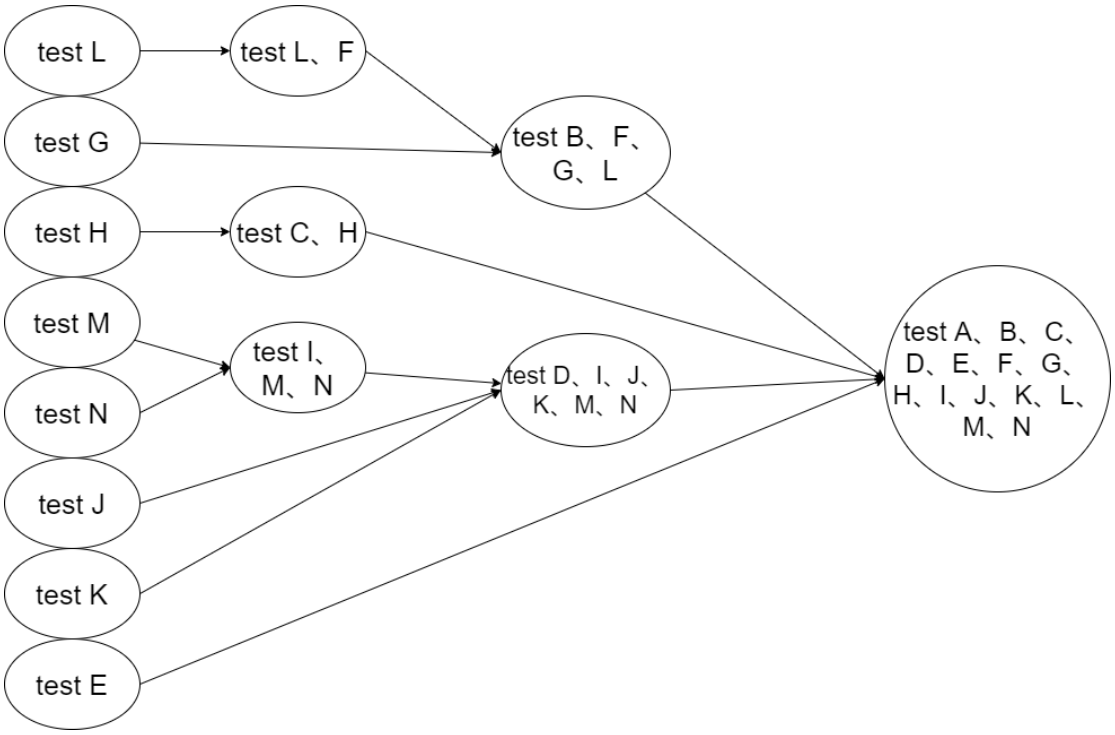
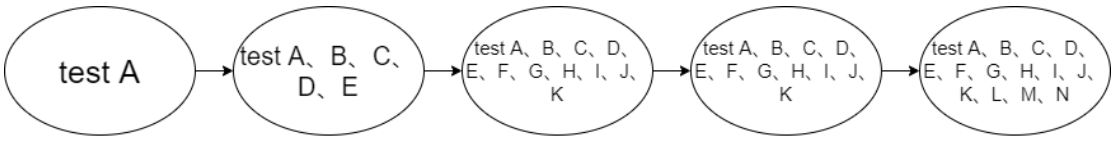


FIGURE 8.22 Example component hierarchy.

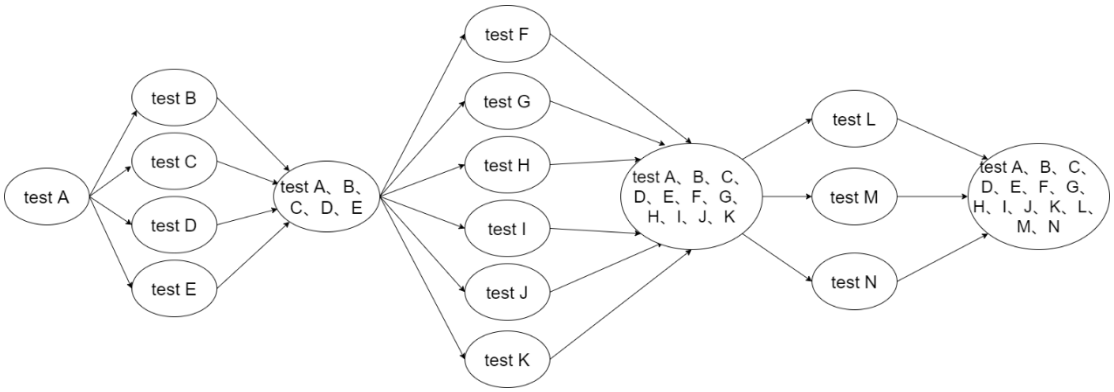
Bottom-up:



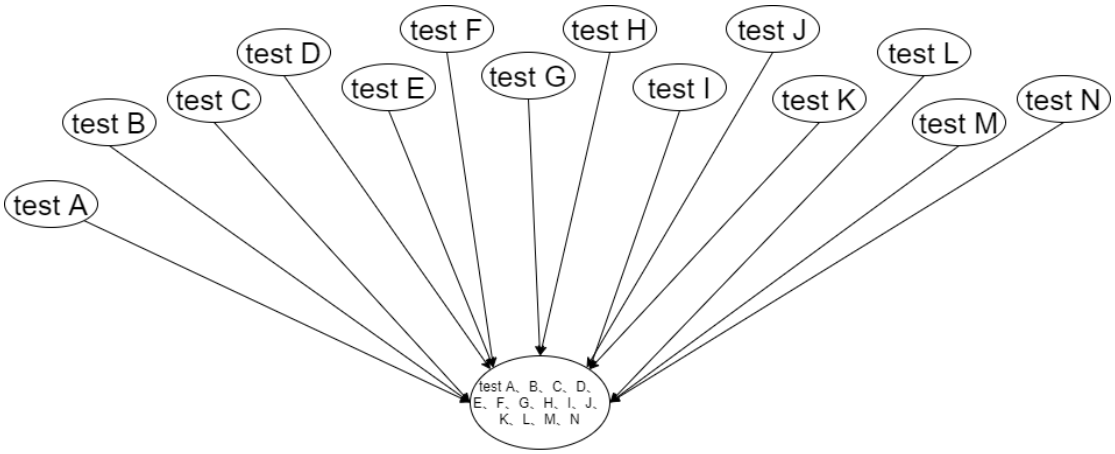
Top-down：



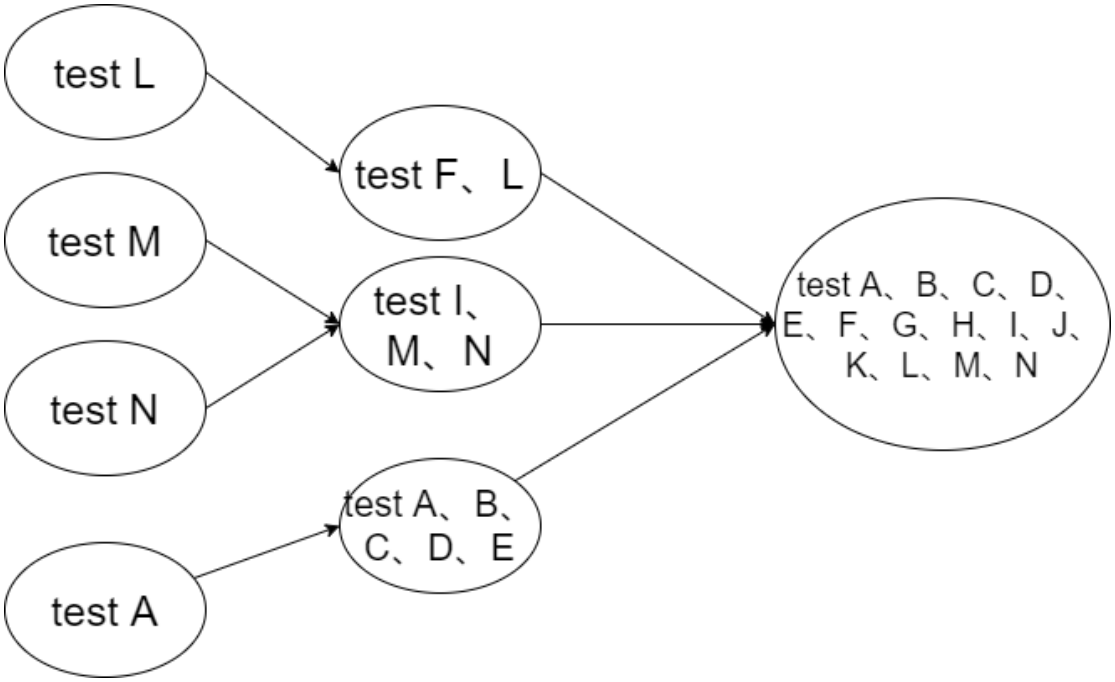
Modified top-down:



Big-bang:



Sandwich:



Modified Sandwich: