

Question 1

Software has been described as being abstract and lacking physical constraints. Briefly explain what these terms imply.

Week 1, Intro 7/8

intangible, no natural limits to the potential of software, changing needs/requirements for users, high risk of failure

Question 2

Outline each phase of software development life cycle

Week 2 SPM 4 Planning->Defining->Designing->Building->Testing->Deployment

Question 3

What are Functional and Non-functional requirements? When considering Non-functional requirements what do

- i) Usability
- ii) Reliability
- iii) Performance mean

Week 1 RA 4

Requirements define the expected services of the system (service statement) constitute functional requirements

Requirements define constraints the system must obey are non functional requirements

Week 1 RA 17

- Usability: define the ease of use of system. A system is more usable if it is easy to use (documentation, help facilities, training, etc.)
- Reliability: the frequency of system failure, correct output and function, system recovery from failure, availability of the system during its operational hours, dependable system
- Performance: the response time of the system, efficient use of resources

Question 4

In the context of Requirements gathering who are the Stakeholders? Briefly describe three types of project that have a different focus with regard to the stakeholders? Give two problems that can occur when using natural language to create a requirements specification.

Week2 RequirementsAnalysis p26-

May involve end-users, managers, engineers involved in maintenance, domain experts, trade unions, etc. These are called stakeholders.

Question 5

In Use Case modelling define the following terms:

- (i) Actor
- (ii) Association
- (iii) Include
- (iv) Extend
- (v) Generalization

Pdf-analysis model- p9-19

An actor specifies a role played by a user or any other system that interacts with the subject.

An association between an actor and a use case indicates that the actor and the use case somehow interact or communicate with each other.

Use case include is a directed relationship between two use cases which is used to show that behavior of the included use case (the addition) is inserted into the behavior of the including (the base) use case.

The include relationship could be used:

to simplify large use case by splitting it into several use cases,

to extract common parts of the behaviors of two or more use cases.

Extend is a directed relationship that specifies how and when the behavior defined in usually supplementary (optional) extending use case can be inserted into the behavior defined in the extended use case.

Generalization is used when you find two or more use cases that have commonalities in behavior, structure, and purpose.

Question 6

Explain the operation of the Airport system described in the Use Case diagram below

图不清晰，叙述参考 week1,Requirement_Analysis2017 p42

Example- Heathcare Patient MS Scenarios 的叙述

Question 7

Draw a simple Use Case scenario for renting a DVD given the following requirements:

- 1) A new user needs to register first, and after they receive a member card
- 2) An existing user can login to the system
- 3) A user shall search the DVD by name, by type, or by actors
- 4) A user shall select the DVD they want to rent to make a reservation
- 5) The system shall send a confirmation email to the user after placing a renting order

week3 Analysis model 2017 p13

Then, explain two of its advantages and disadvantages

Advantages(选两个): p26

Question 8

One well-known Software Process model is the V&V. With the aid of a diagram, describe its sequence of activities. Then, explain two of its advantages and disadvantages? What type of software project does it suit best?

首先要花一幅图

Week1 SoftwareProcessModel_2017 p17

Question 9

Explain, with the aid of a diagram, the Evolutionary Prototyping SDLC model. Give one advantage and one disadvantage of the Prototyping approach.

Pdf- SoftwareProcessModel-p37-39

Question 10

Draw a simple sequence diagram that illustrates the flow of messages that go with the process of making a phone call. There are four participants: the caller, the phone Network, the receiver, and the phone Account

Pdf-Analysis Model_Behavoir modelling-p13(比较像, 但是要改变一些)

Question 11

What are the characteristics of OCL? Pdf-OCL-lecture1-p19

Explain the meaning of the following OCL expressions.

1. $Student.allInstances() \rightarrow forAll(s1, s2 | s1 \neq s2 \text{ implies not } (s1.id = s2.id))$
2. $Set\{1 \dots 100\} \rightarrow iterate\{i: Integer; s: Integer = 0 \mid s + 1\}$

1. Pdf-OCL-lecture2-P30

If the expression that all instances s1 and s2, s1 not equal to s2 implies s1.id not equal to s2.id, then the whole expression is true.

2. Pdf-OCL-lecture2-P34

$S = 1 + 1 + \dots + 1$ (加 100 次)

Question 12

Describe the SCRUM software process model. Ensure to include in your explanation the role of the Scrum Master, Product Owner and the Scrum team. Also, what is the purpose of the meetings at the end of a Sprint?

Pdf-AgileProcess-P30-37

Question 13

Describe three principles of User Interface design. Give one advantage and one disadvantage of each of the interaction styles:

- (i) Menu selection
- (ii) Form fill-in
- (iii) Natural language

Principles: Userfamiliarity/User friendly, Consistency, User guidance, Reduce the user's memory load, Recoverability(forgiveness) ---User Interface Design.pdf p9-10

- (i) Adv. Avoid user errors/ Little typing required

- DisAd. Slow for experienced users/Can become complex if many menu options
- (ii) Ad. Simple data entry/ Easy to learn
DisAd. Take up a lot of screen space
- (iii) Ad. Accessible to casual users/Easily extended
DisAd. Require more typing /Natural language processing systems are unreliable
--- *User Interface Design.pdf* p20

Question 14

What are the main principles for designing user interface(UI)?

Principles: Userfamiliarity/User friendly, Consistency, User guidance, Reduce the user's memory load, Recoverability(forgiveness) ---*User Interface Design.pdf* p9-10 (与 question13 重复)

Question 15

On Architectural Pattern is known as Model-View-Controller(MVC). Explain what this mean, give one advantage and disadvantage to it, and provide a simple example to illustrate its implementation.

Pdf-Week7-ArchitecturalDesign-P6-7

Question 16

In terms of design models, what is software refactoring? Why refactoring is so important in software design?

DesignModel.pdf p26

A reorganization technique that simplifies the design of a component without changing its function and behavior.

Can reduce: redundancy, unused design elements, inefficient or unnecessary algorithms, poorly constructed or inappropriate data structures, or any other design failure that can be corrected to yield a better design.

Question 17

What are the main components in a UML state chart?

Analysis model _ Behavior modelling.pdf p41

State components :

Name describes object state

Action on internal event (on event/action): the action is carried out at the time of occurrence of an event which does not lead to another state

Do action (do / action) : recurring or significant action, carried out in the state

Entry/Exit action (entry/ action), (exit / action) : action is executed instantly as of entry/exit state

Question 18

In the field of Software Testing what are

- (i) Unit testing
- (ii) Integration testing
- (iii) System testing
- (iv) Acceptance testing

Software testing _Software maintenance.pdf p3-4

- (i) An individual unit of software is tested to ensure that it works correctly.

- (ii) Two or more units are tested to ensure that they interoperate correctly.
- (iii) The entire software system is tested to make sure that it works correctly and that it meets/solves the user's needs/problem.
- (iv) The entire software system is tested at the customer environment (field testing or alpha and beta testing) to make sure that it meets the user's needs.

Question 19

What do the terms *Black-Box* and *White-Box* testing mean? Explain two differences between them?

Software testing_Software maintenance.pdf p9

Black-Box: examines the functionality of an application/software under the test without peering into its internal structures or workings

White-box: tests internal structures or workings of an application, as opposed to its functionality

Differences: 综合分析上面的, 然后 White-box 用于 Unit testing, Black-box 用于 Integration testing, system testing

Question 20

What are the characteristics of agile software development?

AgileProcess.pdf p7

Light Weighted methodology

Small to medium sized teams

Vague and/or changing requirements

Vague and/or changing techniques

Simple design

Minimal system into production