Linköpings Universitet
IDA
Kristian Sandahl

2022-08-25

Written exam for Software Engineering Theory

Course codes TDDC88, TDDC93, 725G64

Note: When we visit the exam, I will pass by all students, so you don't need to sit with your hand raised. Just call for my attention when we pass your desk.

If you get trouble with Wiseflow that you or the exam invigilator cannot solve in 10 minutes, call the examiner and continue with other problems.

Instructions to students, please read carefully

- **Explicitly forbidden aids:** Textbooks, machine-written pages, photocopied pages, pages of different format than A4, other electronic equipment than the computer for running Wiseflow. A silent keyboard, mouse and a webcam might be brought.
- Try to solve as many problems as possible.
- Motivate all solutions.
- Please, draw clearly.
- You may write solutions in either Swedish or English, but don't mix in the solution of the same problem.
- Please, note that the problems are not necessarily written in order of difficulty.
- TIP! Read all exercises in the beginning of the exam. This will give you the possibility to ask questions about all parts of the exam since the examiner will visit you in the beginning of the exam time.

Grading

The exam consists of two parts: Fundamental and Advanced.

The Fundamental part has problems worth 10 credits per area. Areas are Requirements, Planning & Processes, Design & Architecture, Testing & SCM, and Software Quality. Thus, the Fundamental part can give maximally 50 credits.

The Advanced part has problems worth 50 credits in total. Each problem typically requires a solution of several pages.

The maximum number of credits assigned to each problem is given within parentheses at the end of the last paragraph of the problem.

Pass condition: At least 4 credits per area in the Fundamental part **and** at least 50 credits in total. The total amount of credits also includes the bonus credits you might have got in lecture exercises autumn 2021. This gives you the mark 3. If you have at least 4 credits for 4 of the areas in the Fundamental part, then you can still pass if you have more than 60 credits in total.

Higher marks are given based on fulfilled *pass condition* **and** higher amounts of credits according to the following table:

Total credits	Mark
0-49	U (no pass)
50-66	3
67-83	4
84-	5

Multiple-choice questions

In multiple-choice questions, we will ask you to write down the letters A, B, C, or D for the one or two statements that you think are true. Note that you should not write down the statements that you think are false. There are exactly two true statements per question, so answering with three or four alternatives gives 0 credits.

For each statement that you select that is correct (i.e., that the statement is, in fact, true) you get one credit. For each statement that you select that is incorrect (i.e., that the statement is, in fact, false, but you believed it was true) you get minus one credit. Each multiple-choice question can give a maximum of 2 credits and minimum 0 credits, i.e., you cannot get negative credits for one multiple choice question.

Example 1: Assume that you have written down statements A and C. If now statements A and B were true, and statements C and D were false, you would get +1 credit for writing down A, but -1 credit for writing down C. Hence, the total credits for the multiple-choice question is 0.

Example 2: Assume that you have written down statement B. If now statement A and B were true, and statements C and D were false, you would get +1 credit for the multiple-choice question.

Example 3: Assume you correctly wrote both statements A and B. If now statements A and B were true, and statements C and D were false, you would get +1 credit for writing down A, and +1 for writing down B. Hence, the total credits for the multiple-choice question is 2.

Good Luck
Kristian

Problems

Part 1: Fundamental

Area 1: Requirements

1 a) Which of the following statements are true? Answer with the statement letter only, no motivation is needed. (2)

A. The IEEE standard 830 is focused on handling *requirements* of memory management.
B. In the *analysis phase* of *requirements engineering*, you use interviews and observation techniques to determine the true needs of the customer.
C. *Software reviews* can be used to *validate* a *requirements specification*.
D. *Traceability* information of a *requirement* can be links to *the classes in the design* that participate in the realization of the requirement

1 b) 1 b) Scenario: You are designing a system for a gym where data about members and their progress is stored. The personal data of the members are used by both members themselves, administrators, and the receptionist. The data comprises name, address, phone, membership category, membership validity, payment options, etc. The progress data is used by members and the personal trainers. Typically, progress data consist of types of exercises, workload, number or repetitions, training record, personal goals etc.

Task: Write at least two *use-cases*, with at least three different *actors* of the gym system introduced above. Draw a *UML use-case diagram* of the use-cases. (4)

Take a photo of the diagram and add that as an appendix to the answer. Let the name of the appendix start with the characters “1b”.

1 c). Write two *user stories* of the system described in **1 b).** Don’t forget to include all properties of a user story. (4)

Area 2: Design and Architecture

2 a) Which of the following statements are true? Answer with the statement letter only, no motivation is needed. (2)

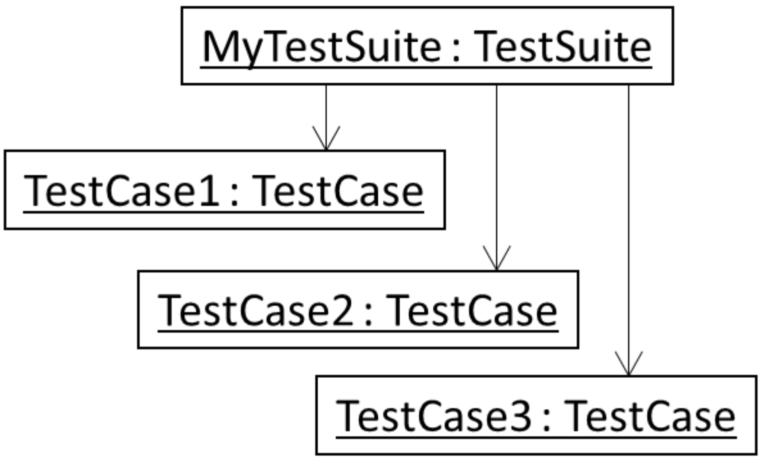
A. The *decompose-and-compose* design approach does not give any benefits if you do not intend to use an *object-oriented language*.
B. The *decompose-and-compose* design approach prescribes that you first attain *coupling* and then *cohesion*.
C. Low *coupling* of a design makes it easier to isolate a change or a *fault*.
D. High *cohesion* of a design makes it easier to understand what a software component is doing.

2 b). Describe a way that you can use to design a *safety-critical* system. Clearly motivate how your suggestion can contribute to safety in 2-4 sentences. (4)

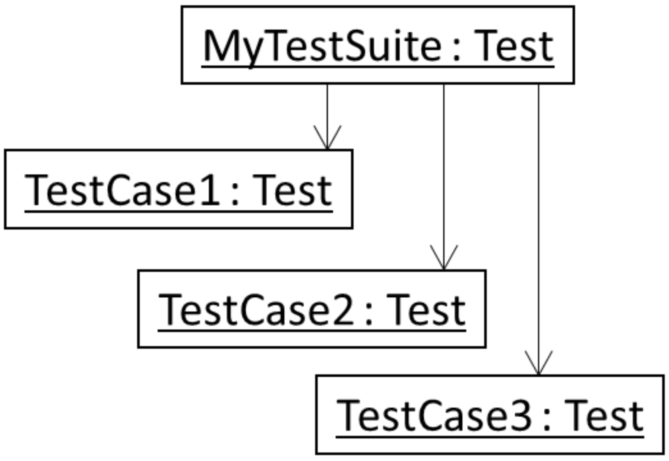
2 c) Draw a *UML class diagram* that allows the *instantiations* (also known as object diagrams) labled A and B, but **not** C. See next page. (4)

Take a photo of the diagram and add that as an appendix to the answer. Let the name of the appendix start with the characters "2c".

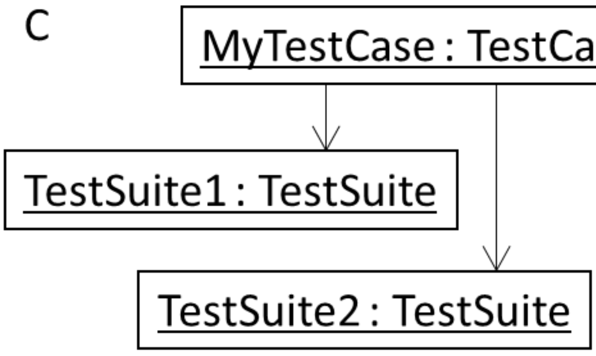
A


MyTestSuite : TestSuite
TestCase1 : TestCase
TestCase2 : TestCase
TestCase3 : TestCase

B


MyTestSuite : Test
TestCase1 : Test
TestCase2 : Test
TestCase3 : Test

C


MyTestCase : TestCase
TestSuite1 : TestSuite
TestSuite2 : TestSuite

Area 3: Testing and SCM

3 a) Which of the following statements are true? Answer with the statement letter only, no motivation is needed. (2)

A. In *Equivalence class testing* you cover one, and only one, *invalid Equivalence class* per *test case*.
B. With *Boundary value testing* you can more easily detect typos of inequality operators in the code.
C. *Equivalence class testing* applies only to *system* and *acceptance testing*.
D. In *Boundary value testing* you don't need to identify *equivalence classes* of *input variables*.

3 b) Somewhere in the process you need to take the decision to stop testing and stop adding new test cases. Describe two different approaches for this. For each approach motivate shortly why the approach is good. 1-2 sentences per motivation. (4)

3 c) Describe how a *feature branch workflow* is organized. Also describe one advantage and one potential disadvantage with this workflow. (4)

If you want to, you can make an illustration of the workflow. In that case, take a photo of the diagram and add that as an appendix to the answer. Let the name of the appendix start with the characters "3c".

Area 4: Planning and Processes

4 a) Which of the following statements are true? Answer with the statement letter only, no motivation is needed. (2)

A. Using the *classical waterfall model* will let you spend more time on testing compared to an *iterative model*.
B. The *classical waterfall model* has more similarities to the life cycle of hardware design than an *iterative model*.
C. Using an *iterative model* is the preferred model for *fixed-price* contracts.
D. When using an *incremental model,* you build the system by implementing and testing parts of the system in small steps.

4 b) Briefly describe four items that you would typically include in *the project organization* part of a *project plan*. One sentence description and one sentence motivation per item is sufficient (4)

4 c) Describe the *Delphi method* for effort estimation. Describe a potential limitation of the method. (4)

Area 5: Software Quality

5 a) Which of the following statements are true? Answer with the statement letter only, no motivation is needed. (2)

A. The duty of the *inspectors* in an *inspection process* is to resolve as many *defects* as possible found in the inspected artifact.
B. The *inspection leader (moderator)* can act as a *recorder* in an *inspection process*.
C. The *author* does not report known or recently found bugs in an inspection; they are handled separately.
D. The *reader* keeps the pace of the *inspection meeting* to ensure that people are not stuck in details.

5 b) Describe with an example the concept of a *CMMI Process area.* (4)

5 c) Describe two *metrics* or *measurements* that can help you to control the *usability* of a product developed in an *iterative* fashion. It's important that the *metrics* can be applied on *prototypes* of different fidelity and maturity through the project. (4)

Part 2: Advanced

6. *Scenario:* Oh, no! After the election in 2022 Sweden got a prime minister who likes to post drastic and provocative messages on Facebook. This creates lot of unnecessary conflicts that aren't even appreciated within the prime minister's own party.

Luckily you have hacked the prime minister's account and you have planted a softbot that alters the formulations or even choice of subject of the messages so that they are aligned with a more normal debate climate. But, the softbot has to be careful of not overusing its capacity, because then its existence might be revealed. The softbot has a partner that watches the debate in the press and gives a debate index 1-5, where 1 is very critical and 5 is very positive about the government.

The specification of the softbot is:

1) Never change the first message of the day.
2) After three messages have been posted in a single day, start altering the formulations.
3) If seven or more messages have been posted during a single day, start altering both the formulations and subjects.
4) If both the formulations and subjects are altered and the debate index becomes larger than or equal to four, stop altering subjects.
5) At 24:00 every night, the system is reset, and no alterations are made

Task: Draw a *UML state diagram* showing the behavior of the softbot**.** (10)

Take a photo of the diagram and add that as an appendix to the answer. Let the name of the appendix start with the character "6".

7. The same scenario as in problem **6**, but the task is to analyze the *input variables* and create *valid equivalence classes*. Make a *test table*, with one *test case* per valid *equivalence class*. (10)

8. Below you will find a list of concepts used in *SCRUM*. For each of the concepts, write down: a definition, expected benefits, potential problems, and a motivation for why or why not the concept can be beneficial to use also in a project using the *classical Water-fall model*.

a) Sprint
b) Product owner
c) Product backlog
d) Burn-down chart
e) Retrospective meeting

(20)

9 Describe the processes of *risk analysis* and *risk planning* in a fair detail. About 6-7 sentences for each of the process.

Also motivate why we should aim for few and project specific risks.

(10)