Linköpings Universitet 2016-01-08
IDA
Kristian Sandahl

Written exam for Software Engineering Theory

Course codes TDDC88, TDDC93, 725G64

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

Instructions to students, please read carefully

- **Explicitly forbidden aids:** Textbooks, machine-written pages, photocopied pages, pages of different format than A4, electronic equipment.
- Try to solve as many problems as possible.
- Motivate all solutions.
- Please, write and draw clearly.
- Write solutions for different areas (fundamental part) and different problems (advanced part) on separate sheets of paper.
- Label all papers with AID-number, date of examination, course code, examination code, and page number.
- You may write solutions in either Swedish or English.
- Please, note that the problems are not necessarily written in order of difficulty.
- TIP! Read through 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 longer 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 2015. 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 with 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 maximum 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 statement and statement C and D were false, you would get +1 credit for the multiple choice question.

Example 3: Assume you correctly wrote both statement A and B. If now statement A and B were true, and statement and 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. Showing stakeholders a prototype of the system is a technique to perform *requirements elicitation*.
B. Showing stakeholders a prototype of the system is a technique to perform *requirements validation*.
C. Acceptance testing is a technique to perform *requirements analysis*.
D. Process adherence is a technique to perform *requirements specification*.

1 b) *Scenario*: You are developing an information system for an employment service. The system is web based and both prospective employees and employers can publish advertisements. Prospective employees can search and apply for jobs directly on the web, and subscribe to newly entered advertisements matching their competence profiles. To use the service, the users need to register contact details to get a free account. The system administrators make sure that old advertisements and unserious advertisements are continually removed.

Task: Write at least two *use-cases* of the system with at least two different *actors*. Draw a *UML use-case diagram* of the *use-cases* and *actors* you created. (4)

1 c) Write two *functional* and two *non-functional requirements* of a competence profile editor that can be used in the system described in the scenario of problem **1 b)**. Clearly mark which requirements you consider being *functional* and *non-functional.* *(4)*

Area 2: Planning and Processes

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

A. A *product backlog* in SCRUM are all historic versions of the product.
B. The *burndown chart* in SCRUM keeps track on which employees that plan to be present at work for the coming week.
C. The *daily SCRUM* is a meeting for the developer team where problems can be brought up.
D. SCRUM is a *process framework* that can be used for different types of projects.

2 b) Describe the process of the *Delphi method* for *effort estimation* in software development. (4)

2c). (4) Describe two advantages and two potential problems of an *iterative* development method.

Area 3: Design and Architecture

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

A. The *deployment view* of the architecture is used by developers to find the code files to work on.
B. The term *provided interface* in a *UML component diagram* refers to an optional function the component can be enhanced with provided that a certain condition is met.
C. The *UML stereotype* <<artifact>> can represent a file of source code.
D. The *execution view* of an architecture can show the flow of data and/or control between components of a system.

3 b) Give an example of a system where both *client-server* and *layered architectural style* can be used simultaneously. Motivate why the use of the architectural styles are beneficiary. (4)

3 c). Draw a *UML object diagram* of four members of your family (or a fictional family, if you prefer). Also draw a *UML class diagram*, which allows the instantiation of the object diagram. (4)

Area 4: Testing and SCM

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

A. An advantage of using a *top-down integration testing strategy* is that you do not need to implement *stubs*.
B. An advantage of using a *big-bang integration testing strategy* is that you do not need to implement *drivers*.
C. A disadvantage of using a *top-down integration testing strategy* is that you postpone the opportunity to discover performance problems of lower-level components.
D. An advantage of using a *bottom-up integration testing strategy* is that you can discover problems with the overall design early in the process.

4 b) Draw a flow chart of a procedure where the minimum number of test cases for *statement coverage* is strictly smaller than the minimum number of test cases for *branch coverage*. Do not forget to motivate your answer. You may use an example. (4)

4 c) Describe the concepts *regression testing, smoke testing, beta testing,* and *performance testing*. (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 concept *non-personal software* derived from *total quality management* means that the software is embedded and not directly accessed by a human user.
B. The principle *mutually beneficial supplier relationships* of *ISO 900-3* implies that you can take steps to help the supplier improve its processes.
C. An organization at CMMI level 1 does not have a risk management process.
D. An organization at CMMI level 3 typically has its standard processes that are adapted to the individual projects.

5 b) Suggest two different *software metrics* that can indicate if the system under development will be easy to maintain. Use the following properties (used in the slides) to describe the metrics: (4)

- Description
- How to obtain data
- How to calculate the metric
- Relevant quality factor

5 c) Describe 4 *roles* that participate in a *software inspection* process. For **each** role describe: i) What they do, and ii) What their responsibilities are. (4)

Part 2: Advanced

6. There are many potential readers of *architecture* documentation. For each of the roles in the list below, describe how they can use the architecture to help them in their work. “Get an overview of the system” is always true, so that does not count. (10)

a) Requirement analyst
b) Designer
c) Tester
d) Project manager
e) Quality coordinator

7. *Scenario*: You are testing a prototype of an almost finished of a website intended for all people who can read and type. To perform the testing you have hired a panel of representative users, and you are now thinking about which data you will collect to give a picture about the *usability* of the system

Task:

a) Describe two types of data that are suitable to capture automatically when the user is using the system. No human intervention shall be needed to gather the data. How would you be able to say something about usability from the data? Are there any potential problems of using the data for decisions about usability?
b) Describe two types of data that are suitable to capture by observing the user using the system. You are not supposed to talk to the user during the usage. How would you like to record the data? How would you like to process the data after the test session? How would you be able to say something about usability from the data?
c) Describe two questions suitable for a questionnaire to be filled by the users after the testing session. How would you be able to say something about usability from the data? Are there any potential problems of using the data for decisions about usability?

Your answers will be judged according to the clarity of your description, but also according to the suitability of the method. If you, for instance, make extremely complex procedures to automatically collect data that is best to collect by observation you will not get full credits. (20)

8. *Scenario:* Lot of things happen when a written exam is given. Some of them are: First, the schedule unit suggests dates and times for the different exams. The result is presented in Time Edit and the examiners are invited to review the schedule of their courses via mail. Comments to the schedule are entered in Time Edit before the deadline. There are no OK-messages sent; if the deadline is passed, the schedule is assumed correct. Issues are resolved by the schedule unit.

The student register via the student portal for the exam and the result is stored in a system called TAL. The scheduler reads TAL and plans rooms and personnel; the result is stored in TAL.

After the exam, the examiner enters the results in exam.liu and confirms the result by sending a mail to the administrator. The administrator prints the official grade list, and when it is singed by the examiner, the administrator sends the result from exam.liu to LADOK. A message about the grade is automatically published on the student portal.

If a student is not registered for the course, the administrator sends a mail to the student, who register via the student portal to LADOK and replies to the administrator when ready. The administrator prints an official grade list for the individual student and enters the result manually in LADOK as soon as the examiner has signed the official grade list. A message is automatically published on the student portal.

Task:

Draw a *UML Sequence diagram* for the system described above. There are many *actors* and *nodes* interacting, so turn the paper in landscape position. Use at least one *fragment*. You do not need to use special format for the arrows of the different types of messages. (10)

9. *Scenario:* Some of the course requirements for getting Master of Science Exam in Computer Science

1. course requirements for a total of 120 ECTS credits from courses from the curriculum
2. passed the course TDDD89
3. courses on advancement level A (advanced) 90 ECTS credits including:
    a. at least 30 ECTS credits courses from the major subject of Computer Science.
    b. a 30 ECTS credits Master's Thesis in the major subject of Computer Science.

Suppose you have written a script that automatically reads applicable data from LADOK, validates the rules above, and either grant or deny the exam. In the case of denial, the number of the failing rule(s) is printed (e.g. 3a)

Task: Create a *test table* of at least five test-cases for *equivalence class testing* of your script given the rules above. You may assume that credits are non-negative integers. Add a column of applicable equivalence class(es) for each test case in the test table. (10)