Linköpings Universitet
IDA
Kristian Sandahl

2020-01-08

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 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 2019. 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 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 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

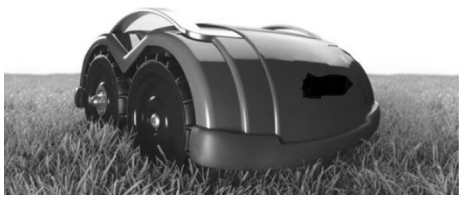
Area 1: Requirements

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

A. Interviewing stakeholders is a generally applicable technique for *requirements elicitation*.
B. *UMLclass* diagrams are mandated content of an IEEE 830 Standard software requirements specification.
C. *Use-case modelling* makes major functions understandable even for non-computer professionals.
D. Classification of requirements is unnecessary if all requirements are equally important.

1 b) Draw a *use-case diagram* for a web-based SCRUM task board. Your *use-case diagram* shall contain two different *actors* and two different *use-cases*.
Hints: Don't forget the use-case description (texts). Use full sentences. Very small operations, such as, logging into the system is not a use-case of its own. (4)

1 c) Write down two *functional* and two well-specified *non-functional* requirements of the control logic of a robot lawn mower. It cuts the grass with various heights moving randomly within an area fended with an underground conductor. When it is time to reload batteries, it finds its own way to the loading station. There are many options for setting preferences. (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. In a *UML state machine diagram* the *transitions* are denoted *action/event*, where *action* is the function to execute during transition, and *event* is the underlying cause for the transition.
B. In a *three-tiered client-server architecture,* you have the possibility to balance the load between servers.
C. The *Façade design pattern* allows algorithms to be selected at runtime.
D. The *implementation view* of an architecture gives you information of, for example, how the source code files of a system are organized

2 b) Describe two different things that you would typically find in an *architecture notebook*. Motivate why they are important to document.

We will not give credits to "boiler plate" content, such as purpose of the document, index, author list, revision history. (4)

2 c) Describe with an example all the following four concepts of an *UML class diagram*: *Attributes*, *operations*, *composition*, and *generalization*. (4)

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 the software engineering terminology an *error* is a human mistake leading to a *fault* in the program, which if executed can cause a *failure*.
B. *Usability testing* very often involves using a panel of test users performing representative tasks of the system.
C. If you want to achieve full *branch coverage testing* of a program the minimal number of test cases is smaller than the minimal number of test cases you would need if you require full *statement coverage testing*.
D. A drawback of *boundary value testing* is that only valid input is tested.

3 b) *Scenario:* With a KLM economy class ticket standard you can check in one bag without a fee at Linköping airport. The maximum weight is 23 kg and the sum of height, width and depth can be 158 cm or lower. There is an app developed where users can enter weight, height, width, and depth of their bag. If the baggage is allowed the app displays a text in blue: "baggage is allowed" and adds the information to your check-in details. If the baggage is not allowed the app displays a text in red "baggage is oversized".

Task: Your task is to identify *equivalence classes* of the input parameters to the app. You shall also make a *table of test cases* that test all equivalence classes.
Hint: You may assume that the input pad for the app only allows entry of digits. (4)

3 c) Describe the goals and workflows of the practices *continuous integration* and *continuous delivery* as being taught in the course. A goal is typically 1-2 sentences. The workflow can be shown in a diagram and/or a numbered list. (4)

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. Activities on the *critical path* must be done on time; otherwise, the entire project will be delayed.
B. A useful property of a *Gantt chart* is that you can mark the current date in the diagram and get a good picture of the progress so far.
C. The *available time* for a task can be calculated as the difference between *float time* and *slack time*.
D. A *burn-down chart* as used in SCRUM keeps track of the financial expenditure of the project.

4 b) Describe the four different *risk planning* activities as taught in the course: *risk avoidance*, *risk transfer*, *risk mitigation*, and making a *contingency plan*. (4)

4 c) Describe three drawbacks and one advantage of *the classical waterfall model*. Don't forget to motivate your answer. (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 Acronym PDCA as used in the Shewhart Cycle means Program, Deploy, Compile, and Analyze.
B. Total Quality Management means that all quality factors are equally important, and must be treated in a similar way.
C. In ISO 9000-3 it is important not to press suppliers too hard to get the lowest prices. They need to keep some room for innovation to the benefit of both suppliers and buyers.
D. In a *usage-based* view of product quality, there can be different opinions about quality, which means that you need statistics do make an objective statement of the quality.

5 b) Describe an activity of the *inspection leader (moderator*) in each of the inspection phases: *Plan and overview*; *Individual checking*; *Inspection meeting*; and *Edit and follow-up*. (4)

5 c) Suppose that you work in a company at *CMMI level* 1 that has high requirements on *usability* of the products. Select the two most important *process areas* of *CMMI level* 2 or 3, that you would like to focus on first. Briefly describe the process areas and motivate how they can contribute to high *usability*. (4)

Part 2: Advanced

6. Describe **four** of the following roles in terms of responsibilities and required competence:

- Project manager
- Product manager
- Lead analyst
- Architect
- Development manager
- Test leader
- Quality coordinator

(Competence is described in terms of skills and experience. For example, “A configuration manager has a sense for good order and has experience from programming and testing”. Avoid trivial constructs such as, “A configuration manager shall be good in managing configurations”.
(10)

7. *Scenario:* You are in charge of designing a “continuous examination” system of a future university. The idea is that students shall be able to test their knowledge in different *subjects*, smaller than a course, whenever they feel prepared.

The student enters a small office with an interactive workstation using different interaction media. By using biometrics and other sensors, the system makes sure that there is only one person in the room and that the identity of the student is known. The student then logs in and navigates to the subject, for instance, Requirements Engineering of the TDDC93 course. The student interacts with the system for a specified time period and answers questions about theoretical concepts and solves sample practical problems. The tasks are sampled from the examiner’s database and most of the examination is graded automatically with advanced classification software. Some parts are fed to teachers for grading in a structured way that makes the grading fast. The results from subject examination are accumulated and when a course is complete, the result is sent to the future LADOK system for inspection and registration. The number of tasks in the database is limited and the students have a limited number of trials, since there are requirements on variation amongst the examination tasks.

Task:

Make a list of important roles of the system, such as clients, servers, databases, and users. Use this list to create a *UML sequence diagram* covering most of the system functions described in the scenario. At least five roles and ten messages are required for full credits. You shall also use some kind of *fragment*. It is allowed to create a solution with several diagrams. (10)

8. *Scenario*: The same as in problem 7.

Tasks:

a) A system like this requires much complex software of high quality. The system is developed gradually over several iterations to minimize the risks. Define the goal for the *first* iteration in terms of what parts (parts can be functions or sub-systems) that are most important to develop and test. Don't forget to motivate the answer. (5)

b) Identify a function in the system and create a table of at least four test-cases for that function. The test cases are intended for automated testing. Motivate how your test cases can be automated. (5)

9. Describe four different metrics that can be used to assess the *usability* of an interactive software product. For each of the metrics answer the following questions:

a) What do you measure? (Called "Description" in the metrics slides.)
b) What procedure do you need to preform to get the data?
c) What resources do you need to collect the data?
d) How do you calculate the numerical number(s)?
e) How does the metric relate to usability? Use arguments such as: "A high number of <my metric> indicates high usability since …"
(20)