Linköpings Universitet 2016-10-28
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 2016. 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

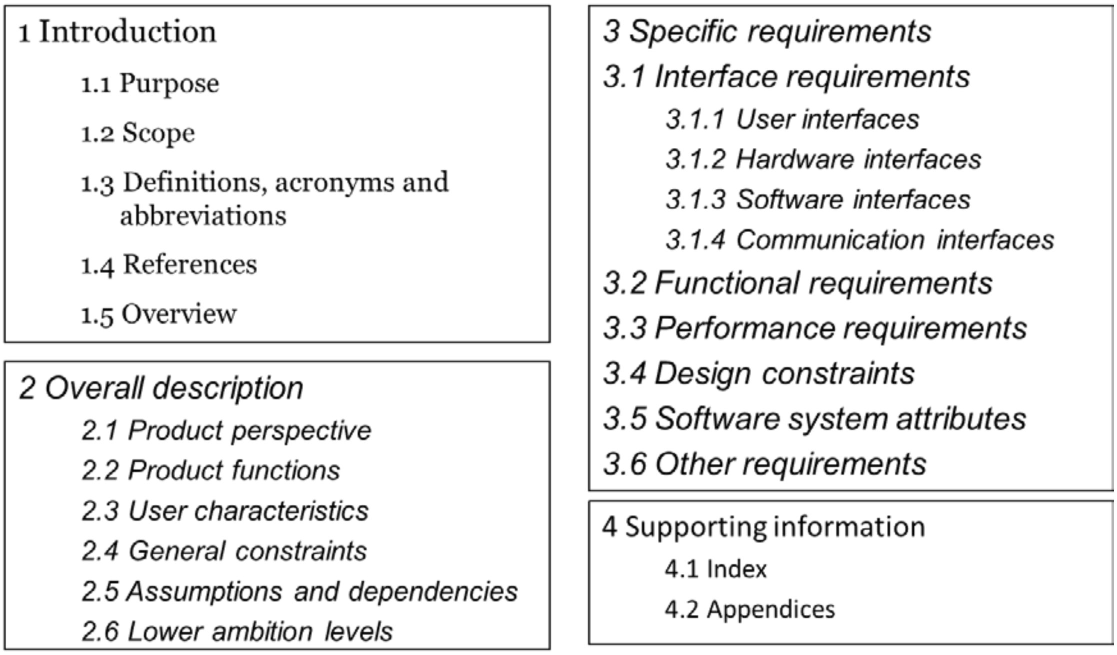
1 a) Which of the following statements are true? Answer with the statement letter only, no motivation is needed. (2)

A. The requirement “When the user presses the OK-button, the system shall upload the file.” is *ambiguous* since it is not certain that a user will ever press the OK-button.
B. The requirement “The system shall be able to upload the file within 60 seconds.” is a *non-functional requirement* since it specifies a quality factor.
C. The requirement “When the user presses the Browse-button, the system shall display the file directory window.” is a *functional requirement*, since it specifies an input-output relationship.
D. The requirement “Many users shall appreciate the commands possible to invoke.” is a *functional requirement*, even though it is not *testable*.

1 b) Scenario: You will design a management system for a car pool. The members of the car pool can sign up for two types of memberships, regular and frequent, with different tariffs for fixed cost and mileage cost. At the member portal, the members can book a car, change bookings, and change personal information. They unlock and lock the car with an app of the mobile phone. If there is a problem with the car, an e-mail is sent to the administrator of the car pool who can decide to send the car to a workshop, maybe by ordering a tow truck. Bookings in conflict with repair time will be automatically changed and members informed. The administrator also maintains a journal of each car and can add new cars to the system.

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) On the next page you will find the headlines of IEEE 830 Standard. What information would you put under headlines 2.1, 2.2, 2.3, and 2.4? Give a short example for each headline to describe the system in **1 b)**. You don’t need to describe the complete system. (4)



1 Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, acronyms and abbreviations

1.4 References

1.5 Overview

2 Overall description

2.1 Product perspective

2.2 Product functions

2.3 User characteristics

2.4 General constraints

2.5 Assumptions and dependencies

2.6 Lower ambition levels

3 Specific requirements

3.1 Interface requirements

3.1.1 User interfaces

3.1.2 Hardware interfaces

3.1.3 Software interfaces

3.1.4 Communication interfaces

3.2 Functional requirements

3.3 Performance requirements

3.4 Design constraints

3.5 Software system attributes

3.6 Other requirements

4 Supporting information

4.1 Index

4.2 Appendices

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. In *Kanban* you must organize the work in *time-boxed sprints*.
B. In *Kanban* you must have an upper limit of the number of items you can work with in parallel.
C. In *SCRUM* you must measure the calendar time it takes from when you started working on an item, until you are finished.
D. In *SCRUM* you must allow the *team* to select items for the *sprint* from the prioritized *product backlog*.

2b) Describe two *stakeholders* of the car management system in problem **1 b)** apart from the member and administrator. Motivate why they are considered as *stakeholders* with 1-2 sentences. (4)

2c). What are the benefits and potential problems of using a development method built on the classic *Waterfall model*? What is the difference between the *classic Waterfall* model and the *V-model*? Don’t forget to motivate your answer.

Hint: Describe four items, for instance, two benefits, one potential problem, and one difference. (4)

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. A *pipe-and-filter* architecture is a way to realize the *Strategy design pattern*.
B. Using a *pipe-and-filter architecture* can imply unnecessary redundancy of the code.
C. *UML diagrams* are classified in *architecture diagrams* and *detailed-design diagrams*.
D. Using a *layered architecture* can imply performance problems.

3 b). Draw a *UML class diagram* that fulfills the following properties of a library management system. Use at least one *generalization* relation and one *composition* relation. You don't need detailed *attributes* and *operations*, just enough to solve the problem.

- The library stores books and magazines.
- For some of the books, the library has several copies.
- The library members can be general members, students, and librarians.
- A general member can have maximally 10 items on loan at the same time.
- A student and a librarian can have maximally 20 items on loan at the same time.
- Each member must have a member card to register the loans in the check-out machine.

(4)

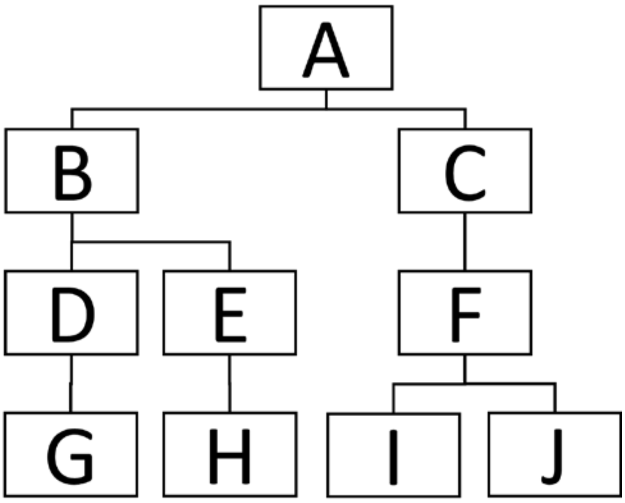
3 c) Suppose you have a *client-server architecture* and you get the feedback from your customers that the system is good, but that the *response time* is too long for everything that the system does. Suggest two different things you can do with the architecture to speed the system up. Don't forget the motivations
Hint: the problem does not specify which variant of client-server architecture you are using, so you have to make an assumption of which one. (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. In a *lock-modify-unlock* version handling system it is not possible to roll the code back to an earlier version.
B. In a *centralized modify-merge* system you update your working copy before *committing* your changes to the server.
C. In a *decentralized modify-merge* version handling system there can be many local *repositories*, but only one public *repository*.
D. In a *decentralized modify-merge* version handling system you send a *pull request* to your co-workers when you have *pushed* your own work.

4 b) Consider the *functional decomposition tree* below. Your task is to describe an integration and testing approach by answering the questions: Which approach will you choose: *top-down* or *bottom-up*? Which test sessions are needed? In what order are they performed? What extra code do you need to write in terms of *stubs* or *drivers*? Why is your selected approach better than the alternative? (4)



4 c) Draw a *flow-graph* of a program where the minimum number of test-cases for *full path coverage* is strictly larger than (>) the minimum number of test-cases for *branch coverage*, which in turn is strictly larger than the minimum number of test-cases for *statement coverage*. (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.** During an *inspection* meeting, all participants try to resolve as many *defects* as possible.
- **B.** All roles in an *inspection* are *inspectors* in the sense that they try to find as many defects as possible.
- **C.** A *technical review* is an informal meeting where system behaviour in different scenarios are reviewed and discussed.
- **D.** A *management review* is focused on following up on plans and budgets.

5 b) *Scenario*: Your newly started company is in the unusual situation that it develops many products from scratch. The customers are happy, but you have to spend lot of time to figure out viable solutions and processes so you wish to be more efficient.

Task: Describe a *CMMI process area* that you think would help you to become more efficient. Summarize the process area's purpose and introductory notes in 5-6 sentences, and motivate how this could help you. (4)

5 c) Explain the measurements *Time To Failure*, *Time To Repair*, and *Time Between Failures*. Give an example of a quality factor that you can estimate with one or more of these measurements as input data. (4)

Part 2: Advanced

6. In this problem you will analyze the prerequisites for two imaginary projects: i) one safety-critical system in the defense industry and ii) one app which can tell the species of birds in a photography.

For the two projects you need to fix two of *the four dependent project parameters* as taught in the course. Which parameters do you choose to fix for project i)? Which parameters do you choose to fix for project ii)? Write a thorough motivation for your choices, which also includes what you can do with the non-fixed parameters if the project comes into trouble.
(10)

7. *Scenario:* You are building a support system for a help desk that customers of broadband services can call if they have any problems. The personnel of the help desk have a one-week education and they use your system, which contains a decision tree and a searchable database of known cases. They can also share their screen with an expert that can be called during daytime. Administrative information about the user, program version, time, etc. are logged.

Task:

a) Write a description of an activity you will use in the development project ensuring good system *usability*. Answer by describing: how the activity is performed, who are involved, what resources you need, when in the project the activity is performed. (8)
b) Describe three different *metrics* that can be used to evaluate *usability*. For each metric, write: a description, how you obtain the data, how you calculate the metric, and how the metric can indicate good or bad *usability*. (12)

8. Identify four different *risks* of the development and/or operation of the helpdesk described in the scenario of problem **7.** by selecting one risk per impact category: *insignificant*, *tolerable*, *serious*, and *catastrophic*. For each of the *risks*, write a short description of the risk and give an example of what you can do if it occurs. (10)

9. Draw a *UML Sequence diagram* of the scenario when the user logs into Facebook and, if granted access, posts a picture from the file system. The user will have the option of entering a picture editing mode, where the user can crop the picture and/or use an automatic white-balance algorithm to improve of the picture. When the user is satisfied the user is prompted with a window to add a describing text from the keyboard. When the user presses return the picture and the text are uploaded. The scenario ends with the user logging out from the system. A web-browser is used to access the Facebook service. Use at least two *fragments*.
(10)