

## 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 200 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. The requirement “The system shall print the current balance of the customer’s account on the receipt.” is a testable requirement given that we have access to the customer account database.
- B. The requirement “The system shall not identify the faults that we normally have.” is an ambiguous requirement.
- C. The requirement “The system shall print the 3 least significant digits of the sum of two automatically generated random integers between 0 and 1000” is testable with black-box testing at system level.
- D. The requirement “The system shall make a prognosis of the weather for the next 24-hour period” is a non-feasible requirement.

**1 b) Scenario:** You are about to develop a commenting system for a newspaper, where it is possible for the general public to comment and debate articles on-line. Before you can send comments you have to register with your true personal data, but you can log in with a nickname. If you break the rules of the debate the editor can erase your comments, and in worst case, expel (ban) you from the system. Any logged in user can press a report-button to catch the editor’s attention of comments using, for example, insulting language.

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

**1 c)** Explain what the following concepts are: *Design constraint*, *Quality requirements*, *Requirements elicitation*, and *Requirements validation*. (4)

Hint: by “explain” we mean that a student like you who has not taken the course shall be able to understand what you wrote. (1-2 sentences per concept)

## 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. An advantage with the *waterfall model* is that you have more testing compared to other models.
- B. An advantage with *iterative models* is that they are well suited for large hardware projects that can be co-designed with the software.
- C. A problem with *iterative models* is that it might be hard to select the requirements that shall be attained in a specific *iteration*
- D. A good thing with the *waterfall model* is that it is simple to understand from a company management perspective.

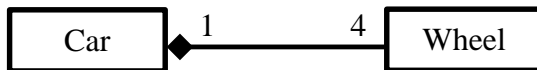
2b) Explain what a *toll-gate* and a *mile-stone are*. (4)

2c) What is the definition of a *stakeholder* of a software project? Give examples of three stakeholders of the student portal at LiU. (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 UML *Class diagram* shows a *static structure* of the designed system.
- B. An *abstract class* can only contain *abstract operations*.
- C. Just writing a “\*” as a *multiplicity* is a short-hand for “1..\*”.
- D. The *composition relation* prohibits sharing, for instance, in the diagram below we cannot shift wheels from one car to another.



3 b) Make a *UML sequence diagram* of a web shop of t-shirts, where you pay with credit card, with at least 4 *nodes* using at least one *fragment*. (4)

3 c) Already when you make the architecture of a system you can attain different quality factors. For instance, if performance is in focus you can design for parallel execution. This question is: How can you make a *software architecture* that is suitable for a *safety-critical* system? Write a short motivation. (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. *Statement coverage* means that you make a set of test-cases that executes all decision branches in a program at least once.
- B. You need at least as many test-cases to reach *full path coverage* compared to *branch coverage*. You might need more test-cases.
- C. The minimum number of test-cases for *statement coverage* of a program is the *cyclomatic complexity*,  $V(G)$ , of the programs' corresponding flow graph.
- D. *Coverage criteria* can be used to determine when to stop testing.

4 b) Define the four concepts: *Error*, *Fault*, *Failure*, and *Oracle*. (4)

4 c) Describe the workflow of *continuous integration*. (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 percentage of commands invoked by test users can be an indication of *usability*; the higher the value the more usable is the software since the high usage indicates that it is fit-for purpose.
- B. An approximation of *reliability* can be calculated by the quotient  $(MTTF+1)/MTTF$ , where MTTF is Mean Time To Failure.
- C. The *depth of the inheritance tree* (DIT) is calculated for object-oriented software as an indication of how hard it is for a human to understand what the software is doing.
- D. Measuring the *non-comment lines of code* is a tedious work that must be done by hand, since it is hard to identify good comments.

5 b) *Scenario*: You feel caught in a trap; your company is doing fairly well, but each time you deliver software on time, customers detect several bugs that cost money to fix. At the same time, whenever you deliver high quality software you are late, which means that you have to compensate your customer for the missing delivery. Your software is not safety critical.

*Task*: Identify a *CMMI process area* that can help you to deliver good software on time. Summarize the process area's purpose and introductory notes in 5-6 sentences, and motivate how this could help you. (4)

5 c) Explain what is done during the four phases of an inspection: *Plan and overview*, *Individual checking*, *Inspection meeting*, and *Edit and follow-up*. (4)

## Part 2: Advanced

6. *Scenario*: You are about to start a development project for a mobile system under the following circumstances:

- The goal is to help disabled people to find their way around in the center of large cities. For example, you shall avoid stairs.
- The Client development team is located in Sweden and the Backend development team is working in India.
- In India you have a data base specialist. She was one of the top students in the country and has worked for your company for 7 years now.
- In Sweden, you have a very good UX (User experience) designer who is working half-time as university researcher and teacher.
- You are planning to use a software called Phone Gap to develop for multiple mobile platforms at the same time. For functions supported the generated code is almost as good as the natively programmed code.
- You might need to use NFC (Near Field Communication) which is hard to use with iPhone, but you have a reliable source that it will be supported in the coming 12 month period.
- Your first deliverable is scheduled in 10 months from now.
- You may make more assumptions, as long as you write them down.

Your task is now to formulate five *risks* and show how they will be managed. Each risk shall contain:

- description,
- short motivation for the *Probability*,
- short motivation for the *Impact*,
- and a suitable, motivated *plan* for the Risk. (10)

7. There are 12 *principles* from the *Agile manifesto*, for this problem we have selected 4 of them:

*Principle I*: Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

*Principle IV*: Business people and developers must work together daily throughout the project.

*Principle VII*: Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

*Principle X*: Simplicity--the art of maximizing the amount of work not done--is essential. (continued on next page)

7. (continued) For each of the principles, you shall write about:
- How you would organize a project practically to follow the principle. We expect a thorough description with at least three different practices of how to create that organization. It is natural that some practices overlap between principles, but the motivation of why the practice contributes to the principle might differ.
  - The major differences to a traditional *waterfall project model*. One difference with a motivation is enough.
  - The potential risks of following the principle? One risk with a motivation is enough. (20)

8. *Scenario:* In a computer game screen there are several action buttons you can click to make progress in the game. If you move the mouse pointer over an action button and keep it within the borders for more than 500 ms, a help text will be shown with an estimate of the probability of success of pressing the action button. If you press the select button on the mouse, the action button in the game is “armed”. If you now release the select button on the mouse while still pointing at the action button, then the action associated with the button of the game is executed. However, if you don’t release the select button on the mouse and instead move outside the “armed” button in the game, the action is not executed.

*Tasks:*

- Draw a *UML State Machine diagram* for the *class* of the action button with the above functionality. Please note that the scenario is not specifying all alternatives that can happen with the mouse and action button so you need to make assumptions that you also write down. (10)
- Create a table of *test cases* for the functions that ensures that all *transitions* in the state diagram in task a) are traversed at least once. (10)

Student’s adventure game

Next action

