

## 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 2014. 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. Interviewing *stakeholders* is a generally applicable technique *for requirements elicitation*.
- B. *Use-case modelling* makes major functions understandable even for non-computer professionals.
- C. The requirement “The project team shall use SCRUM with a sprint length of 30 days” is a *functional requirement*.
- D. Good layout, paper quality and a change-tolerant page-number system are the main reasons to use a *formal method* for *requirements specification*.

1 b) Draw a *UML use-case diagram* of a system for a social network system, such as Facebook or LinkedIn. There shall be two different *use-cases* and two different *actors* in the diagram. Don’t forget the use case texts. Use full sentences. Logging in to the system is not a use-case of its own. (4)

1 c) Explain what the following concepts are: *Feature*, *user story*, *requirements analysis*, and *the IEEE standard 830*. (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. Activities on the *critical path* in a GANNT chart must be done on time; otherwise the entire project will be delayed.
- B. During the *daily SCRUM meetings* all team members brainstorm to find solutions to impediments discovered the day before.
- C. *SCRUM* is a general *process framework* that can be used for other activities than software development.
- D. *Time-boxing* in *SCRUM* means that if not all tasks in a *sprint* are completed, then the sprint’s length is allowed to be extended maximally one more week.

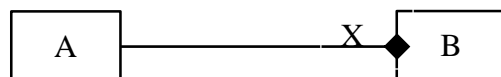
**2b)** Describe four problems of using the *classical waterfall life-cycle* that are not likely to be present using an *agile* development processes. There shall be at least one problem from the customer's perspective and at least one problem from the developers' perspective. For each problem, write a sentence of why it is not likely to appear with *agile* methods. (4)

**2c)** Describe the four *dependent project parameters* as taught in the course. (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. *UML sequence diagrams* are used to describe the static structure of a system.
- B. The *rationale* of an *architecture description document* gives instructions of how to implement the source code in a rational way.
- C. A software module named "FileIO" (abbreviation for File Input Output), containing only functions related to reading and writing contents of files, is an example of a module with high *cohesion*.
- D. In a *UML class diagram* using a *composition relation*, the multiplicity X can only be 0..1 or 1.



**3 b)** Describe two advantages and two disadvantages with a *layered architectural style*. (4)

**3 c)** Draw a *UML class diagram* using at least one *composition* relationship and at least one *generalization* relationship. Instances of the classes in the diagram shall obey the following rules:

- All vehicles must have a registered owner.
- Cars and motorcycles are vehicles.
- A car must have exactly four regular wheels and exactly one spare wheel.
- A motorcycle must either have two or three regular wheels.

(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. A *driver* is a small piece of software replacing an upper layer in integration testing, by calling the module under test with some predefined parameter settings
- B. *Stress test* means that you test what is happening if a user only gets a limited amount of time to try to enter the correct input
- C. *Regression test* means that you use linear regression on data from test-logs to find out when to stop testing.
- D. If you want to achieve *branch coverage testing* of a program you need at least as many test cases as you would need if you were satisfied with *statement coverage testing*.

4 b) Suppose that your system has a *functional decomposition tree* with few levels (not more than 3), but high fan out (7-10 descendants per node). Describe one advantage and one potential risk with applying: i) a *top-down integration testing strategy*, and ii) a *big-bang integration testing strategy*. Don't forget to motivate all your answers. (4)

4c) Explain the input, output, and functions being processed of a *build and integration server* as it is used in *continuous integration*.<sup>1</sup> (4)

## Area 5: Software Quality

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

- A. In a *flow-graph* of a program, that can only contain *binary decisions*, the *cyclomatic complexity*,  $V(G)$ , of the program grows quadratically with the number of decisions as more decisions are added.
- B. The *cyclomatic complexity* has impact both for the *maintenance* and *testability* of the software.
- C. A high *number of reused code lines* can be an indicator of high *reliability* with the argument that the reused parts have been tested and run for a longer time than newly produced code.
- D. *Failure intensity* is the probability that the system will work as intended during a specified time interval.

---

<sup>1</sup> Yes, this is exactly the same question on 4c as October 2014, but we received very few good answers.

**5 b)** Shortly describe the five *maturity levels* of the *staged CMMI* model. Give two different examples of positive things that can be expected for the customers of the system when their vendor seriously starts climbing up in maturity levels. (4)

**5 c)** Explain the roles of *software inspections*: *Inspection leader* (aka *moderator*) and *inspector*. Also, give an example of: an *entry criterion* and an *exit decision* of an *inspection process*. (4)

## Part 2: Advanced

6. A friend of yours wants to introduce *risk management* in his/her company. Therefore, your friend asks you to write a convince-your-boss letter explaining what risk management is. Credits will be given for texts explaining the different *types of risks* and the different *steps in risk management*. Some kind of introduction about the benefits of risk management is needed. (10)

7. 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 perform 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)

8. *Scenario*: When you buy railway tickets on-line you go through the following steps:

1. You specify departure and destination stations.
2. You specify if you need return ticket and the preferred times for outgoing and returning trains.
3. You specify number of passengers and if individual passengers are eligible for discounts.
4. You will get up to five suggestions of travel routes plus prices for different types of tickets. If you have a return ticket you will get similar information for the return. If you are logged in with your frequent traveller id, the customer database is checked for your normal preferences, before presenting alternatives.
5. You can either click the suggestions you want to buy or ask for earlier or later alternatives.
6. You can now check for added services, including seat preferences.
7. You come to a summary of your order and are asked to confirm or go back. You are asked to write down the full name of all passengers.
8. When the order is confirmed you select type of ticket delivery and payment method. You can either make a bank withdrawal or pay with credit card. Depending on the choice you are directed to your on-line bank or the credit card payment.

When everything is OK, you will be shown a receipt and a link to an e-ticket PDF-file if that option was selected

*Tasks*:

- a) Draw a *UML class diagram* of the concepts being handled by the system. Don't forget to write name and *multiplicity* of the *associations*. (10)
- b) Draw a *UML sequence diagram* with at least four nodes describing this process. Hint: use the different types of *fragments* and *guard conditions* (10)