

## Written exam for Software Engineering Theory

Course codes TDDC88, TDDC93, 725G64

**Note: There are more than 200 students writing the exam. When I come to your room, I will walk through the all the rows of tables. You don't need to sit with a raised hand all the time. Catch my attention when I pass your table.**

### ***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 2012. 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 statement 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. We say that two requirements are *testable* if, and only if, they can always be satisfied simultaneously.
- B. *Prototyping* can be used both for *requirements elicitation and requirements validation*.
- C. An *Entity-Relationship diagram* is useful when we want to describe the dynamic behaviour of an embedded control system.
- D. The IEEE Standard 830 for Software Requirements Specification encourages you to adapt the disposition of headlines to your particular application.

1 b) A centralized patient information management system keeps all information of patients treated by the hospitals in a health care region. Health care personnel and the patients themselves can read or sometimes edit the information. There are strict rules for *confidentiality, integrity and availability*. Draw a *use-case diagram* of this system with two different use-cases and two different actors. 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) Write down two *functional* and two *non-functional* requirements of the patient information system described in problem 1b). (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. The process in SCRUM is the main focus, so the *sprint retrospective* continues as long as is necessary even if it delays the *sprint planning*.
- B. A *task board* in SCRUM is an organization that sets priorities on changed and new *product backlog items*.
- C. SCRUM can be used for many different types of projects, not only software development.
- D. Many practices from eXtreme Programming (XP), such as *pair programming, test-first programming, and refactoring* can be chosen to be used by a SCRUM team without violating ideas behind SCRUM.

2 b) Describe the four *dependent project parameters* as taught in the course. (4)

2 c) Describe three advantages and one drawback of *iterative software development*. (4)

### Area 3: Design and Architecture

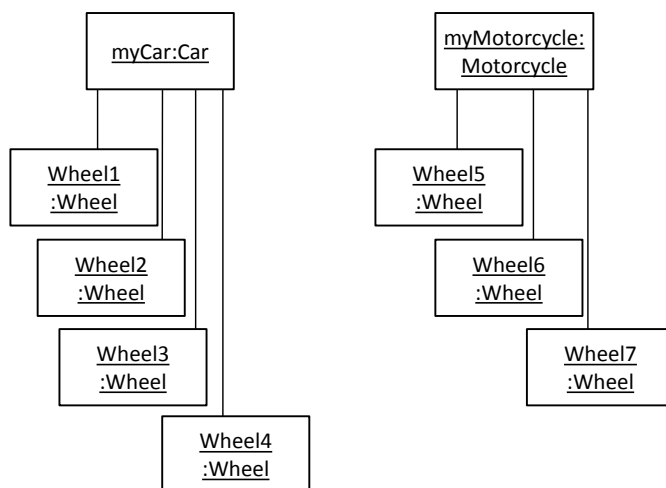
3 a) Which of the following statements are true? Answer with the statement letter only, no motivation is needed. (2)

- A. In a *UML state diagram* the *transitions* are denoted *action/event*, where *action* is the functions 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 *Facade 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.

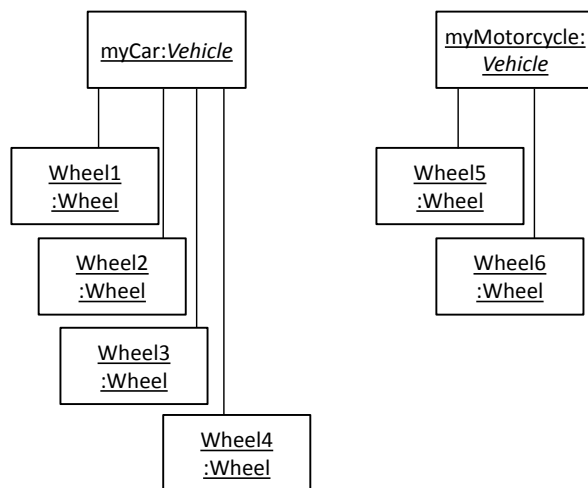
3 b) Describe three advantages of a *layered architecture style*. Describe the concept of *layer bridging*. (4)

3 c) Draw a *class diagram* of the classes Vehicle, Car, Motorcycle, and Wheel that allows the instantiations in I., but not the ones in II., and not those in III: (4).

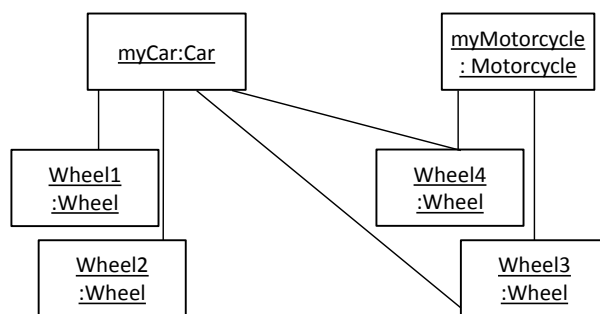
I:



II:



III:



## 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. *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.
- B. 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*.
- C. 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*.
- D. *Regression test* means that you use linear regression on data from test-logs to find out when to stop testing.

**4 b)** Write down an advantage of each of the integration testing strategies: *Big-bang*, *Top-down*, *Bottom-up*, and *Sandwich integration testing*. (4)

**4 c)** Describe the work-flow of *continuous integration* from the perspective of a developer. (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 object-oriented software, a class's *depth in the inheritance tree* means that the software is harder to understand. A large depth gives lower *understandability*. The reason is that in a deep hierarchy, you are more likely to find *inherited methods and attributes*.
- B. 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.
- C. If we draw a *flow-graph* of a program, that only contains *binary decisions*, then the *cyclomatic complexity*,  $V(G)$ , of the program grows quadratically with the number of decisions.
- D. If  $MTBF = \text{Mean Time Between Failures}$ , the *Reliability* can be approximated by  $MTBF/(1-MTBF)$

**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)** It is often said that people working in an organization at *CMMI level 3* needs to be more creative than on lower levels. Describe two *process areas* at CMMI level 3 that gives an example of when creativity is needed. Don't forget to motivate the answer. (4)

## Part 2: Advanced

6. A home temperature control system has the following requirements:
  1. The cooler and the heater can never be on simultaneously.
  2. If the cooler is on and the temperature becomes 18 °C or lower, then the cooler is turned off.
  3. If the heater is on and the temperature becomes 22 °C or higher, then the heater is turned off.
  4. If the both the heater and the cooler have been off for 20 minutes or more and the temperature is 18 °C or lower, then the heater is turned on.
  5. If both the heater and the cooler have been off for 20 minutes or more and the temperature becomes 22 °C, then the cooler is turned on.
  - a) Assume that you are the test leader at the company developing the control system. Identify the *input parameters* and *equivalence classes*. Motivate your choice. (5)
  - b) Create a test table of five test cases performing *boundary value testing*. Motivate why you selected the test cases you did. (10)
  - c) Describe the advantages and disadvantages of *boundary value testing* compared to randomly generating test cases. (5)
  
7. 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.
  9. When everything is OK, you will be shown a receipt and a link to an e-ticket PDF-file if that option was selected.

Your task is now to draw a *UML sequence chart* with at least four nodes describing this process. Hint: use the different types of *fragments* and *guard conditions*. (10)

8. A friend of yours wants to introduce *risk management* at his company. Since he knows about your great talent, he asks you to write him a convince-your-boss letter explaining what risk management is: the different types of risks, the different steps in risk management, the benefits you will get from risk management, documents needed, meetings needed, practical hints etc. (20)