

## Written exam for Software Engineering Theory

Course codes TDDC88, TDDC93, 725G64

**Note: There are more than 200 students in the exam. When I come to your room, I will walk through 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 2013. 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) Consider the following requirements of an interactive project-planning tool:

**R1:** The user shall be able to add a new task to the project.

**R2:** The critical path of the Gantt chart shall be displayed in red.

**R3:** The colorizing according to **R2** shall be done maximally 2.0 seconds counted from the last keystroke creating or modifying the critical path.

**R4:** When a change in the critical path is made, the tasks that no longer are on the critical path shall be colorized in their default colour after the new critical path has become red.

**R5:** When a change in the critical path is made, the tasks that no longer are on the critical path shall be colorized in their default colour before the new critical path has become red.

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

- A. R1 is a *functional requirement*
- B. R2 is a *non-functional requirement*
- C. R3 is a *non-functional requirement*
- D. R4 and R5 are *consistent*

1 b) *Scenario:* Your hairdresser salon has started a web-site for frequent customers. If you have signed up for the free membership you can login and book treatments. You can search for different employees, date, or time. The hairdressers use the system for planning, for instance, entering working hours and serving drop-in customers. The customer can erase bookings if more than 24 hours remain before the start of the treatment. The hairdressers can edit anything, and can also edit advertisements from their vendors.

*Task:* Now, create a *use-case diagram* of the web-site consisting of two different actors and two different use-cases. Don't forget the use case texts. Only logging in and logging out are basic functions, not to be considered as use-cases. (4)

1 c) Describe two techniques for performing *requirements elicitation*. Describe two different goals for *requirements analysis*. (4)

Hint: by "describe" we mean that a student like you but who has not taken the course shall be able to understand what you wrote.

## 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)

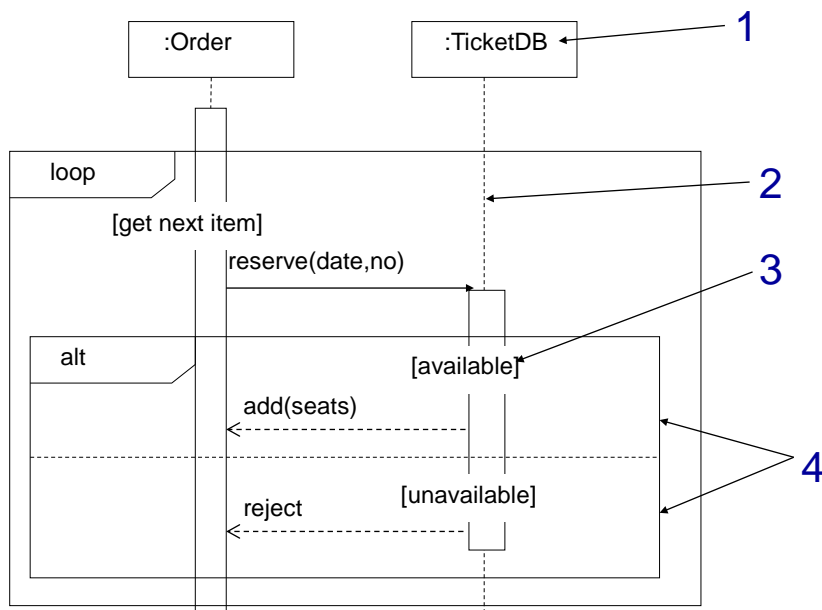
- A. Linköping University is a good example of a *project*, since we have a well elaborated *strategy*.
- B. According to Frederick Brooks it is advisable to have a risk list of about  $\frac{n(n-1)}{2}$  risks in a team with  $n$  members.
- C. The dependent project parameters of a project are: *Resources*, *Features*, *Calendar time*, and *Quality*.
- D. In a *Gantt chart* a *task* and its *predecessor* do not need to belong to the same *phase*.

2b) Describe two advantages of working with a *Waterfall life-cycle model* and two advantages of working with *Iterative development methods*. (4)

2c) Describe the four tasks performed in the *risk management process*. (4)  
Hint: don't mix this with the *strategies*: avoidance, transfer, and acceptance.

## Area 3: Design and Architecture

3 a) Consider the following excerpt from a sequence diagram.



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

- A. Arrow number 1 is pointing to a *class*.
- B. Arrow number 2 is pointing to a *dependency association*.
- C. Arrow number 3 is pointing to a *guard condition*, which means that this region is entered only if the guard condition is true.
- D. Arrows labelled with number 4 is pointing to two *alternate branches*.

**3 b)** Draw an architecture of a web-shop with a line-and-box diagram. Point out the following elements of the diagram: *module*, *relationship*, and *interface*. Which architectural view is your diagram showing? Don't forget to motivate your answer to the last question. (4)

Hint: Don't spend too much time in making the architecture perfect, 5-10 elements will do.

**3 c)** Define the two concepts *pipe* and *filter*. Give one advantage and one disadvantage of the *pipe-and-filter architectural style*. (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. If you want to start integration testing with high-level user interface functions you will need several *drivers* to emulate lower level functions.
- B. *Bottom-up integration testing* can help you to discover performance problems of lower level components early in the testing process.
- C. With *big-bang integration testing* there is a high risk that localizing the fault of a failure will take time compared to other strategies.
- D. *Sandwich strategies* start integrating and testing modules at a *target level*, continue with a *bottom-up method* to the highest level, and then with a *top-down* method to integrate the lowest level modules.

**4 b) Scenario:** With a SAS economy class ticket you can pay EUR 80 to check in second bag. 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 calls a page for making a reservation. 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. (4)

Hint: You may assume that the input pad for the app only allows entry of digits.

**4c) Scenario:** Adam is working on the file `TestCase.java`. He gets his copy from the *code repository* and edits the method `getInput`. While he is working, Eva also gets her copy of `TestCase.java`, edits the same method, and writes back the new code to the repository before Adam is finished. Poor Adam now has two revisions of `TestCase.java`: His own and the one in the repository. Adam and Eva are using a *Centralized Modify-Merge* version handling tool, for instance, Subversion.

*Task:* Your task is to describe a workflow for Adam that starts with him getting the copy of the file and ends with both his and Eva's changes of the `testCase.java` stored in the repository. Use the correct commands instead of the everyday language in the scenario. (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. When you have a requirement on *reliability* you often express the requirement in terms of *failure intensity* since this is more straight-forward to measure instead of estimating a probability.
- B. The number of hours that your team puts on testing can be said to be a measurement of the structure of your code.
- C. McCabe suggested in his article about *cyclomatic complexity* that all code modules shall have a *cyclomatic* complexity of 10 or higher. Otherwise there is a risk that people make errors of omission by forgetting conditions.
- D. *Efficiency* of interactive software can be measured by logging the time it takes for representative users to complete a specified task.

**5 b)** Two questions on Software Reviews: i) What is the difference between an *inspection* and an *audit*? ii) Apart from finding faults, the *author* of the inspected document has two responsibilities in the inspection process. Describe these two shortly. (4)

**5 c) Scenario:** Life isn't easy: Your end-users complain that even though your software is failure-free and has a good looking GUI, you miss certain functions that "everyone" in their company know are sometimes needed. Your employees are really appreciating your personal feed-back, but it comes sporadically and is based on your feelings only. As a complement they would like to have more regular and objective feed-back.

*Task:* Your task is now to find the two most relevant *CMMI process areas* that can help you to take care of the criticism. Write down their names and a short motivation. (4)

## Part 2: Advanced

6. Make a template of headlines of a *project plan*. For each headline, add a small instruction within brackets of what information that goes under that headline.

Example:

### **1. Project description**

#### **1.1 Background to the project**

< Major functions. Identify orderer and main users. Expected benefits from the perspective of orderer. Earlier solutions used. >

#### **1.2 Constraints**

<...>

(end example)

You shall assume that requirements specification and quality management are documented in separate documents. At least 10 headlines are needed for full credits. 1.1 in the example does not count; 1.2 in the example does if you write instructions. (10)

7. Make a comparative analysis of the three agile frameworks. SCRUM, Kanban, and eXtreme Programming (XP). For each of the frameworks write:

- A short description of what developers and other stakeholders can expect from the framework. 2-3 sentences
- A list of important concepts of the framework, for instance, roles, artefacts, meetings, and technical practices. For each concept, make a 1-2 sentence description and give a short motivation if you think this concept is unique to the framework or if there is a similar concept in another framework.

For full credits you describe at least two concepts per framework and 7 concepts in total. (20)

8. *Scenario:* You are developing a global peace stabilization system (GPSS). The idea with the system is to collect lot of information about potential conflicts in the world. The more different evidence you have, the stronger is the degree of alert:

1. **Green:** the region is observed due to some incidents.
2. **Yellow:** there are conflicts in the region.
3. **Red:** immediate attention by the United Nations is required.

You have several sources classified as:

- **Indicators**, for instance, changes in the argumentation in media.
- **Reliable**, for instance, video clips taken by habitants.
- **Highly reliable**, for instance, satellite photos.

Each source has its own data collection method, data base, and completely automated data analysis. The output from each source is the percentage of conflict likelihood and a threshold value, indicating that conflicts are detected.

The rules for the alerts are:

- Green: two indicators pass threshold value, or a reliable source is 10 per cent units below its threshold.
- Yellow: two reliable sources pass the threshold, or one highly reliable source is 10 per cent units below threshold.
- Red: one highly reliable source passes the threshold and two reliable sources pass the threshold.
- If the conditions for the current alert level are changed, the level is lowered to the level where the conditions are fulfilled.

*Task:* Your task is now:

a) Show with an example *UML class diagram* how the *strategy design pattern* can be used to model the decision-making software, called Black-board, and its relation with the sources. Don't forget that the diagram probably needs a small explaining text. If you have forgotten the design pattern, you can get some credits for a general UML class diagram. (10)

b) Make a *UML State chart* for a class named `AlertLevel` that holds the current level of alert. The rules specified above shall be obeyed. Don't forget to document assumptions made. (10)

Hint: you need a no-alert state.