Written exam for Software Engineering Theory

Date: 2007-06-05  
Time: 14.00-18.00

Valid for pass of the written exam in courses: TDDB61, TDDB62, TDDC01, TDDC06, TDDC88, TDDC93.

Allowed aids: Two sheets of handwritten A4 pages. You may write on both pages, with any type of size and colours. One volume of dictionary to or from English or an English wordbook.

Explicitly forbidden aids: Textbooks, machine-written pages, photocopied pages, pages of different format than A4, electronic equipment.

Graded exams will be handed out 2007-08-15 in Donald Knuth between 12.00 and 13.00.

Questions of clarification will be answered by assistant Andreas Borg, who will visit the exam about an hour after start. Examiner Kristian Sandahl can be reached on phone 0706-68 19 57 during the exam.

Instructions to students, please read carefully

- Try to solve as many problems as possible.
- Motivate all solutions.
- Please, write and draw clearly.
- Write only on one side of the paper.
- Write solutions to problems on separate sheets of paper.
- Label all papers with name and your Swedish personal number.
- You may write solutions in either Swedish or English.
- Please, note that the problems are not necessarily written in order of difficulty.
**Grading**

The exam consists of two parts: Fundamental and Advanced.

The Fundamental part has problems worth 10 credits per area. Areas are: Requirements, Design & Architecture, Testing, Planning & Processes, and Quality factors. Thus the Fundamental part can give maximally 50 credits.

The Advanced part has problems with 50 credits in total. They can be distributed over two to five problems. Each problem typically requires a longer solution of several pages.

The maximum number of credits assigned to each problem is given in within parentheses at the end of the last paragraph of the problem.

**Pass condition:** At least 6 credits per area in the Fundamental part and at least 50 credits in total. This gives you the mark 3 in the Swedish system and an E in ECTS.

Higher marks are given based on fulfilled pass condition and higher amounts of credits according to the following table:

<table>
<thead>
<tr>
<th>Total credits</th>
<th>Mark in Swedish system</th>
<th>Mark in ECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>100-85</td>
<td>5</td>
<td>A</td>
</tr>
<tr>
<td>84-76</td>
<td>4</td>
<td>B</td>
</tr>
<tr>
<td>75-67</td>
<td>4</td>
<td>C</td>
</tr>
<tr>
<td>66-58</td>
<td>3</td>
<td>D</td>
</tr>
<tr>
<td>57-50</td>
<td>3</td>
<td>E</td>
</tr>
<tr>
<td>49-0</td>
<td>UK</td>
<td>F</td>
</tr>
</tbody>
</table>

**Good Luck!**

**Kristian**
Problems

Part 1: Fundamental

Area 1: Requirements
1. Suggest one primary, one secondary, and one indirect stakeholder\(^1\) of a university student records system. Make sure to motivate the answer. (3)

2. Suppose you are specifying a system reading signals from a sensor. From the NASA experience we learned that it is important to specify the units of signals. Now, give two other properties about a sensor signal that are worth mentioning in a requirements specification. (2)

3. Give one advantage and one disadvantage of specifying requirements in natural language only. (2)

4. Write and draw a UML diagram of a use-case of a system administrating time reports in a consultancy firm. (3)

Area 2: Design and Architecture
5. Give two arguments of why it is good to have high cohesion and low coupling amongst software modules. (2)

6. Draw a valid object diagram of at least five objects that can be instantiated from the class diagram below. (3)

```
Person

Teacher 1 teaches 1..250 Student
```

7. Give two advantages of using a pipe-and-filter architecture. (2)

8. Explain how the Strategy design pattern works. (3)

---

\(^1\) Some keywords that have a special meaning in Software Engineering are written in italic font.
**Area 3: Testing**

9. Explain the concept of *daily build.* (3)

10. Explain the concept *coverage criteria* in white-box testing. (2)

11. Consider a system with the following component hierarchy

```
          A
         / \   \
        B   C  D
       / \ / \ /
      F  G H  I
     / \ / \  /
    J   K L   M
```

Write down a diagram showing the sequence and dependency for a *top-down integration testing* of the system. Is this a good strategy for the system? Write down a motivation for your decision. (5)

**Area 4: Planning and Processes**

12. Give two problems that can be encountered by using the *incremental* life-cycle model. (2)

13. Write down a process of at least 5 steps that can be used to make a *risk analysis* of a software development project. (3)

14. Describe a *prototyping* life-cycle model. Be sure to mention draw-backs and advantages. (5)

**Area 5: Quality factors**

15. Give at least one example of how the customer can earn money by buying software from a company with well documented development processes. You may assume that the processes are followed by developers, who continuously give process improvement suggestions. (2)

16. Define the *reliability* and *availability* of a system. (2)

17. Suppose you are consultant of an immature company. Suggest three things the company can do to climb towards a certification at *CMM or CMMI level 2.* (3)

18. Suggest three different measures of the size of an *object-oriented* system. (3)
Part 2: Advanced

19. Describe at least five principles or properties of agile methods. If you know of a certain agile method you can describe that instead. Describe a quality factor that you think is well attained by using an agile method. Motivate your answer thoroughly. Are there any special prerequisites of the product or your organisation that have to be met to give the expected benefits? (10)

20. Write a UML state chart of a cruise control system of a car (Swedish: farthållare). Use at least five states with specified transitions. Some requirements:
   1. The system does not work under 50 km/h.
   2. If the throttle pedal is pressed while cruise control maintains a user selected speed, the car will accelerate. If the throttle pedal is released again, the cruise control system resumes maintaining the user selected speed.
   3. If the break pedal or transmission pedal is pressed the cruise control system is turned off. But it remembers the latest user selected speed, and uses that as default speed if the user turns the cruise control system on again.
   4. If the engine is turned off or the handbrake is engaged, the cruise control system is off and the latest user selected speed is forgotten.
   5. While the cruise control system is maintaining the speed of the car, the user can increase or decrease the user selected speed in steps of 1 km/h by clicking either of two buttons (+ and -)

You may invent new requirements or modify the existing ones if you document your changes. (20)

21. One of your vendors sends you a mail where the requirements specification of one of your competitors is mistakenly attached. What do you do? Motivate your answer. (5)

22. Describe three different architectural styles. For each of the styles, discuss the applicability to two different problems:
   1. A web-based data collection system, where teachers record the progress of students’ lab series results. Students can watch their results, and reports can be generated for the administrator.
   2. An alarm function in a wrist-watch. (15)