Written exam for Software Engineering Theory

Date: 2007-10-22
Time: 08.00-12.00

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 color. One volume of dictionary to/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 2007-11-07 in Donald Knuth (IDA) between 12.30 and 14.00.

Questions of clarification will be answered by Kristian Sandahl, who will visit the exam about one hour after start. Examiner David Broman can be reached on phone 0707-909075 during the exam.

Instruktioner till tentamensvakter
Studenterna får ha med sig 2 handskrivna A4-blad med text på båda sidorna och ett lexikon. Studenter med andra hjälpmedel utan särskilt tillstånd får inte påbörja tentamen förrän examinator kontaktats.

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 for different areas (fundamental part) and different problems (advanced part) 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 worth 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.

If you passed the Quiz exam 2007-09-20, 10 extra credits will be automatically added to your result of the exam. You do not need to state if you passed the quiz exam or not.

**Pass condition:** At least 5 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. Note that if credits are added for passing the quiz exam, it will not affect the requirement of at least 5 credits per area in the Fundamental part.

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>
</tr>
</thead>
<tbody>
<tr>
<td>100-84</td>
<td>5</td>
</tr>
<tr>
<td>83-67</td>
<td>4</td>
</tr>
<tr>
<td>66-50</td>
<td>3</td>
</tr>
<tr>
<td>49-0</td>
<td>UK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total credits</th>
<th>Mark in ECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>100-96</td>
<td>A</td>
</tr>
<tr>
<td>95-83</td>
<td>B</td>
</tr>
<tr>
<td>82-68</td>
<td>C</td>
</tr>
<tr>
<td>67-55</td>
<td>D</td>
</tr>
<tr>
<td>54-50</td>
<td>E</td>
</tr>
<tr>
<td>49-0</td>
<td>F</td>
</tr>
</tbody>
</table>

**Good Luck!**

**David & Kristian**
Problems

Part 1: Fundamental

Area 1: Requirements

1a) Suggest three different readers of a requirements specification. Be sure to explain what they will look for in the specification. (4)

1b) Explain the terms actor, use-case and system boundary. (3)

1c) Suggest two ways in which a requirements specification can be validated before the system is coded and tested. Compare the approaches in terms of effort. (3)

Area 2: Design and Architecture

2a) Create an UML activity diagram modeling a student applying for a job describing the following scenario:

The student is writing a letter that is sent to the company. The company registers the application and a manager takes a decision if the student should be interviewed or not. If the student should be interviewed, the manager books a meeting time. In both cases, the student gets an e-mail with a response to the application. If the student is called to the interview, he will get the response if he got the job at this meeting. At the same time as the company manages the application, the student is taking a course at the university. If the student accepts the job he will start at the company after the course is finished.

Divide the diagram into suitable swimlanes and give proper names. (4)

2b) There are three fundamental concepts in security. Which are they and what do they mean? Describe each of the concepts in the context of an on-line banking system. (3)

2c) Assume that you are designing a program that is required to read very large text files and transform these files into different formats. Different kind of transformations should be able to be applied and it should be simple to extend or add new transformations. Which architecture style is suitable for this system? Give at least one advantageous feature with this approach and one drawback. Explain it in the context of the example. (3)
**Area 3: Testing**

3a) Is it necessary to have an oracle when doing performance testing? Motivate your answer. (2)

3b) Describe the concept of *Sandwich integration testing*. Give a small example of at least 10 components. Mention at least one benefit with this approach. (4)

3c) Suggest 2 things you can note in an Inspection Record apart from the defects. (2)

3d) Describe the differences between alpha-testing and beta-testing. (2)

**Area 4: Planning and Processes**

4a) What is a *milestone*? What is the difference compared to a *tollgate*? State at least 2 properties of a well defined milestone. (2)

4b) It is important but hard to estimate the effort of performing different development tasks in a software engineering project. Choose an effort estimation method described in the course literature or at the lecture. Describe the approach. Discuss pros and cons to use the method. (2)

4c) Risk management:
   - After a risk has been identified in a risk management process, it needs to be analyzed. Often two numbers are assigned to the risk in this phase. What do these numbers stand for? (1p)
   - The next step is to make a plan for the risk. Give an example of how a risk can be transferred. (1p)
   - What does it mean to define a contingency plan? Give an example. (1p)

4d) Sync and Stabilize is an approach used by for example Microsoft. Describe this approach and its parts. Argue for why or why not this approach is good software engineering. (3)

**Area 5: Quality factors**

5a) Give a reason why the number of defects per line is normally larger for code modules with many lines-of-code. (2)

5b) Describe the basic ideas behind reliability engineering. (3)

5c) Describe three principles of Total Quality Management. (3)

5d) Is it recommendable to start working with process areas at CMMI-level 4 before attaining all levels at level 3? Motivate your answer. (2)
Part 2: Advanced

1) Imagine that you are a consultant specialized in software engineering processes and models. The development manager Kent of a company SmallSoftware AB asks you for advice on how they should improve their software engineering process. The company produces off-the-shelf software for handling stocks at small production companies. Currently, they always collect requirements, do a large design, implement and then test. However, this approach often fails, since requirements come in late in the cycle, resulting in large design changes and delayed products.

   • a) Since you are a clever consultant, you directly realize that SmallSoftware AB is using a kind of waterfall model. Explain in a detailed and pedagogic way the different stages in the waterfall model, the main principles, and the drawbacks that you see with it. Try to imagine and expand the scenario and put it in the context of SmallSoftware AB. (6)

   • b) You know a lot about two processes called Rational Unified Process (RUP) and Scrum. Argue why you would think that these approaches are suitable for SmallSoftware AB. Discuss which one would be most suitable and why. Make clear assumptions about how the company is working. Do not forget to clearly explain how the processes are working, since Kent has a very brief knowledge about software processes in general. (10)

   • c) Choose 3 quality factors that you think are especially important to SmallSoftware AB. Explain the principles of these quality factors. Discuss how these can be measured and why they can be hard to measure. Reflect and discuss how one of these factors can be affected by the choice of development process (RUP, Waterfall, and Scrum). (6)
2) As the combined analyst and chief software architect of a large air flight company, you are assigned the task to develop a new web-based software system. The CEO of your company has given you some very brief information about what this system should be able to handle in the following e-mail:

Dear Anders,

As you already know, we want to launch this new web-based ticket system within 1 year. The main purpose is that we want to connect our customers more tightly to our organization by offering this good service. I have some short input for the system. First of all, we need to keep track of who the users are, so that we can advertise special offers etc. I also think it would be good to have a bonus program where the customers get points depending on how much they are flying. Then, of course, it must be possible to search and book flights. It must definitely also be possible to buy the tickets using credit cards at the website, but other payment methods should also be possible.

I hope that this gives you an idea of what we in the management team are looking for.

Best regards,
Dennis

- **a)** Create a use-case for the system and write down the scenario text. The scenario should involve at least 2 actors. Draw a use case diagram which includes this use-case + two more use-cases for which you do not need to give any scenario. (Remember that a use-case is not a simple function of a system.) (5)

- **b)** Create a short but good list of requirements for the system. At least 7 functional and 3 non-functional requirements should be stated separately. Please remember all details that should be part of good requirements. (5)

- **c)** Imagine that you are now creating the architecture of the system. Draw a box-and-line diagram of the system. The system should contain at least 4 sub-systems. Do not forget to explain the diagram *in detail*, both regarding purposes of sub-systems, relations, and interfaces. (5)
3) Anna is a professional software developer who decided to develop a software game in her spare time. The game is a pretty advanced 3D game, where the player can view the play field from different angles. She decides to use a famous design pattern for this purpose. Here UML sketch is shown below:

- a) Which design pattern is she trying to use? Explain the main concept of this pattern (2)

- b) Unfortunately, she has made some small modeling mistakes, and therefore the solution does not work. Find and describe the mistakes. How should it be changed? One of the mistakes creates a problem when more views are added. Which one? Why? (6)

- c) Draw the corrected UML diagram and add one more 3D view class, where the game player should be able to view it from a flying camera (call the class View3DFlying). Add an interface named Flyer, which has an operation called flyUp. Force the new class View3DFlying to implement Flyer, by adding a realization relationship. (4)

- d) Would it be a better idea to make View3D an interface? If so, what would the semantic difference then be? (1)