Written exam for Software Engineering Theory

Date: 2009-08-24  
Time: 14:00-18:00

Valid for pass of the written exam in courses: TDDC88 and TDDC93.

Allowed aids: Two sheets of handwritten A4 pages. You may write on both pages, with any type of size and colour. 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 shown 2009-09-09 at 11:30-12:30 in Conference room Donald Knuth

Questions of clarification will be answered by examiner Kristian Sandahl (013-28 19 57 or 0706-68 19 57), who will visit the exam about one hour after start.

Instruktioner till tentamensvakter
Studenterna får ha 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 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 and start with the ones you directly see a solution to. This will also 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, 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.

Multiple choice questions will ask you to write down the label of two correct statements. Credits are given according to the following table:

<table>
<thead>
<tr>
<th>Number of correct statements in the answer</th>
<th>Number of false statements in the answer</th>
<th>Number of credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2 or higher</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1 or higher</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>any</td>
<td>0</td>
</tr>
</tbody>
</table>

**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 a C 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>Translation to ECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>100-84</td>
<td>5</td>
<td>A</td>
</tr>
<tr>
<td>83-67</td>
<td>4</td>
<td>B</td>
</tr>
<tr>
<td>66-50</td>
<td>3</td>
<td>C</td>
</tr>
<tr>
<td>49-0</td>
<td>UK</td>
<td>Fx</td>
</tr>
</tbody>
</table>

**Good Luck!**

**Kristian**
Problems

Part 1: Fundamental

Area 1: Requirements

1 a) Which of the following statements are true? Answer with the statement number only. No motivation is needed. (2)
   1. An actor in a UML use-case diagram can be either a human user or another system.
   2. A single person can represent two different stakeholders in the development of a system.
   3. The requirements “The system response time shall be at least 20 ms.” and “The system response time shall be at least 30 ms” are consistent.
   4. The scope of the system must be determined before requirements elicitation can start.

1 b) Draw a UML Use-case diagram and write Use-case texts for a cinema ticket booking system. There shall be at least 4 use-cases. (4)

1 c) Write down one functional and one non-functional requirement of a mail management client, for instance MS Outlook. Give two reasons of why it is a good idea to classify and separate functional requirements and non-functional requirements in a requirements specification. (4)

Area 2: Design and Architecture

2 a) Which of the following statements are true? Answer with the statement number only. No motivation is needed. (2)
   1. The strategy design pattern uses inheritance and polymorphism of object-oriented design.
   2. Structural design patterns, for instance, Singleton, deal with initializing and configuring of classes and objects.
   3. A reason for striving for low coupling is that future changes can be kept cheap.
   4. A layered architecture emphasises the execution speed of the system.

2 b) Suggest four different things you need to document about an architecture of a system. For each of the four things, give an example of a stakeholder that can benefit from reading this information. For instance: A system overview (documented thing) is good for the new programmer’s (stakeholder) understanding of the documentation. (4)

2 c) Draw a two-state UML statechart of a system of your own choice and mark the following elements in the diagram: State, transition, trigger event and action. (4)
**Area 3: Testing**

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

1) It is not sufficient to know that an input condition specifies the range of numbers, 1-100, for a variable to create equivalence classes for test cases.
2) When you use top-down integration testing you need lot of component drivers.
3) Functional system testing is best performed by a team independent of design and implementation.
4) If a variable is found defined and then killed in data flow analysis you do not have an invalid system, but potentially a programming problem.

3b) Consider the following flowchart:

```
Start

no

X=/=0

Z=Z-X

Z=sin(X)

yes

Z>10

yes

Z=0

no

Z=Z/X

end
```

Write two sets of test cases with different number of test-cases: one set for branch coverage and one set for path coverage. Each test case is specified with values for Z and X. (4)

3c) Explain the terms top-down integration testing, stress test, regression testing and the termination problem in the context of software testing. (4)
**Area 4: Planning and Processes**

4a) Which of the following statements are true? Answer with the statement number only. No motivation is needed. (2)

1. Using an incremental process model reduces the need for configuration management since each increment is smaller than if the whole product was developed with the waterfall model.
2. In his article in 1970, Royce, was himself critical against a strict, one-pass waterfall model.
3. Agile methods, such as XP, emphasise that you develop as many functions as possible to give the user a rich set of alternatives to choose from.
4. Using an incremental development model makes it possible to organize the work in such a way that independent parts of the system can be developed in parallel and thus reduce time-to-market.

4b) Write down a process describing how the Delphi method can be used to estimate the effort needed to implement a set of functions. Give one drawback about the Delphi method. (4)

4c) Describe four different ways of how you can reduce risk that you identified for a project. (4)

**Area 5: Quality factors**

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

1. In experiments it has been possible to show that by specialising the tasks of inspectors, an increase in the defect detection rate can be obtained.
2. The statistical capture-recapture theory predicts that if the independent inspectors have very few defects in common, then it is very likely that they detected almost all defects of the inspected document.
3. A Root-Cause-Analysis process is often supported and documented with a Gantt chart.
4. The moderator in the Graham and Gilb inspection process has the responsibility for follow-up activities after the Inspection Record is written.

5b) Suggest four metrics that can be used to assess the usability of a system. (4)

5c) Describe briefly the characteristics of organisations rated on CMMI levels 1, 2 and 3. Also describe how you can do to climb from level 1 to a higher level. (4)
Part 2: Advanced

6) Describe the following architectural styles:
   • Pipes and filters
   • Layers
   • Process control
   • Blackboard

For each of the styles give a:
1. Brief description of the most important mechanism. An informal box-and-
   arrow diagram with a sentence or two will do.
2. An advantage with the architectural style.
3. A drawback or difficulty with the architectural style.
4. A quality factor that is likely to be well-attained with a system using the
   architectural style. Don’t forget to motivate the answer.
5. A detailed description of a metric that can be used to measure the quality
   factor of your choice. (20)
7) You are designing a software cooperation tool with the following features:

- Project planning system with several diagrams possible
- Resource planning
- Time reporting
- Work item handling, both opening, accepting and closing a work item.
- Version control and configuration management.
- Remote video conferencing.
- Process guidance tool
- Integrated Development Environment
- Etc.

Start sketching the system by doing the following:

- **a)** Draw a use-case diagram of two different actors and three use-cases. Don't forget the use-case texts. Add a fourth use-case that is relevant for one of the actors but not included in the system. Make this visible with a subject field and a system boundary line in a correct way. (5)

- **b)** Draw a class diagram of some part of the system, or a high-level model of the entire system. Use at least 5 classes with association, generalisation, and composition. All classes shall have at least one attribute and one operation. Make sure to get a sensible multiplicity on the associations. (10)

- **c)** Draw a UML state diagram of the class Time report. Use at least three states (apart from the standard begin and end states) and 2 transitions. (5)

- **d)** Draw a UML activity diagram for a change management process. Use at least 5 activities, 1 decision and a fork or synchronisation bar. (5)

- **e)** Write 5 functional test cases for a part of the system with input and expected output. (5)