Linköpings Universitet
IDA
Kristian Sandahl

Written exam for Software Engineering Theory

Course codes TDDC88, TDDC93, 725G64

Note: 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. Two different types of *stakeholders* of a system can have conflicting requirements of a forthcoming system.

    B. The major goal of the *Requirements Abstraction Model (RAM)* is to ensure that specifications are written according to the IEEE Standard 830 for Software Requirements Specification.

    C. We say that a requirement is *feasible* if the requirement is not possible to fulfil with the available personnel and technology.

    D. It can be a good idea to classify requirements into *functional and non-functional* requirements, since these types of requirements are specified and tested in very different ways.

1 b) Draw a *use-case diagram* for a mobile app online shop, such as, Google Play (earlier known as Android Market) or App Store. Your *use-case diagram* shall contain two different *actors* and two different *use-cases*.
Hints: Don't forget the use-case texts. Use full sentences. Very small operations, such as, logging into the system is not a use-case of its own. (4)

1 c) Write down two *functional* and two well-specified *non-functional* requirements of a vending machine accepting coins and/or credit cards. Assume that the machine contains a refrigerator, to sell candy, cakes and soft drinks. (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. During the *daily SCRUM meetings* all team members brainstorm to find solutions to impediments discovered the day before.

    B. One of several similarities between *OpenUP* and *SCRUM* is that they have the same set of *roles*.

    C. *User stories* in *eXtreme Programming* are very short descriptions of what the user wants to do with the system.

    D. The *velocity* of a SCRUM team can be found by analysing the *burn-down charts* of the team.

2 b) Describe the difference between a *phase* and a *task* in a Gantt chart. Also, describe the difference between a *milestone* and a *tollgate*. (4)

2 c) Describe three drawbacks and one advantage of *the classical waterfall model*. (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. A *message* in a *UML sequence diagram* cannot contain parameters, as in a method or function call.
  B. An advantage of a *layered architecture style* is that an implementation of one layer can be replaced without causing changes to the other layers.
  C. The *Strategy design pattern* is a good choice when a change to one object requires changing others.
  D. In a *UML class diagram*, a *dependency relation* from A to B indicates that changes in the definition of B causes changes of the definition of A.
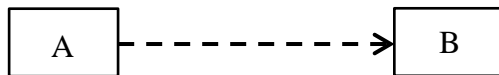
3 b) Describe the concepts *pipe* and *filter* in a *pipe-and-filter architecture*. Describe one advantage and one disadvantage of this *architectural style*. (4)

3 c) Draw a *UML class diagram* using at least one *composition* relationship and at least one *generalization* relationship. Instances of the classes in the diagram shall obey the following rules:
  * All vehicles must have a registered owner.
  * Cars and motorbikes are vehicles.
  * A car must have exactly four regular wheels and exactly one spare wheel.
  * A motorbike must have exactly two regular wheels.
(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. *Smoke tests* consist of a set of selected test cases that can be run fast and is used in *continuous integration*.
  B. *Usability testing* very often involves using a panel of test users performing representative tasks of the system.
  C. When you *commit* a file from a SCM repository, you download the latest version from the repository.
  D. A drawback of *Equivalence class testing* is that only valid input is tested.

4 b) Describe the following steps of the highest levels of testing: *Function test, Performance test, Acceptance test, and Installation test.* Focus on **what** is tested, not **how** testing is done. (4)

4 c) Suppose that your system has a f*unctional decomposition tree* with many levels (more than 5), but low fan out (0-2 descendants per node). Describe one advantage and one potential risk with applying: i) a *top-down integration testing strategy*, and a ii) *bottom-up integration testing strategy.* (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. The primary goal of an *inspection meeting* is to *detect* defects, not to *resolve* them.
  B. The *recorder* of an inspection is responsible for making sure that the *entry criteria* for the *inspection* are met.
  C. It is highly recommended to collect data from the *inspections* in order to improve the costly *inspection process* itself.
  D. In contrast to a *technical review*, an *inspection* can only be performed on finished and complete artefacts.

5 b) Explain how you calculate the *Cyclomatic Complexity, V(G)* of a connected *flow graph* that represents a piece of software. You may use an example with a V(G) value higher than 5. Give a quality factor that can be (partially) predicted with V(G). Don't forget to motivate the answer. (4)

5 c) Describe the following concepts from *CMMI*: *Process area, Risk Management, Verification, and Validation.* (4)

Part 2: Advanced

6. A Car Cruise System (CCS) monitors all cars on special highways. A special feature is to build and automatically drive queues of cars. This means that the driver releases steering and speed control to the CCS, which monitors and drives the cars in a queue with small distances between the cars in a fast, but safe and even speed. When driven in the queue the attention of the driver is checked through biological sensors. If the driver is close to fall asleep, the car is moved out of the queue and parked at the nearest parking lot. All cars in a queue communicate via an ad hoc 4G network, and sends GPS positions to the CCS. There are lot of sensors and actuators to detect and fix potential problems found in the close environment of the car and the queue. A car enters a queue by telling the CCS that its driver is interested in queue driving and the CCS starts inviting cars in the neighborhood. A car can also be invited if it is close to a queue. The driver can press a button to leave the queue, a procedure done in different steps, such as increasing the distance around the car and moving it to another lane before handing off control to the driver.

Now your task is to model the system. You may add assumptions and constraints as long as they are explained.

   a) Draw a *UML class diagram* of the main concepts related above. Some classes are given, such as the Car, Queue, Driver, and CCS. Other concepts might be modeled as *classes* or *attributes*, that is your choice. Don't forget the *multiplicity* and names on *associations*. The solutions are judged for their clarity to display the relations between the most important concepts using a correct UML syntax. Methods are less important and are only added if they increase the clarity of the solution. Use about 8-10 concepts from the text above. (15)
   b) Draw a *UML Statechart* for the class Car. The *statechart* shall consist of at least 5 *states*. Don't forget *triggering events* and *actions* on the transitions. The solutions are judged for clarity, consistency and correct use of *UML Statecharts*. Hint: If it is hard to draw rectangles with rounded corners, use ovals instead. (15)

7. Let us assume that you have taken over a software company with about 30 employees. You are now the CEO of the company and own the majority of the shares, that is, you have total control and responsibility of the company's operation. Your predecessor has been conservative, only taking on short assignments with very clear specifications. The company has mainly worked with different administrative systems, including both adaptations of large systems, and development of web-based systems. The company's reputation is that they deliver on time and within budget, but there has always been a problem with the quality, which means lot of bug-fixing and support after first delivery. You have set your mind to increase the quality, and you are now writing a letter to the employees where you briefly describe and motivate the **three** most important things you intend to start within the company to increase quality. Write down a draft of the letter which clearly presents the three different things you have chosen. Write about the expected benefits, and how you are going to introduce these things in your company. (20)