

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. The goal of *requirements elicitation* is that the requirements engineer gets an understanding of the true needs of the customer.
- B. We say that a requirement is *unambiguous* if it can only be interpreted in one way by different people.
- C. The *UML stereotype* <<extend>> links a *use-case* to the *classes* participating in realizing the use-case.
- D. The IEEE Standard 830 for Software Requirements Specification mandates the use of only *natural language* in requirements descriptions.

1 b) Draw a *use-case diagram* for a web-based SCRUM task board. 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 the control logic of a robot lawn mower. I cuts the grass with various heights randomly within an area feeds with underground conductor, when it is time to reload batteries it finds its own way to the loading station. There are lot of options for setting preferences. (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. The process in SCRUM is the main focus, so the *sprint retrospective* continues as long as is necessary even if it delays the *sprint planning*
- 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. SCRUM is a general process framework that can be used for other activities than software development.

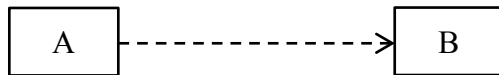
2 b) Describe the four different risk planning activities as taught in the course: *risk avoidance*, *risk transfer*, *risk mitigation*, and making a *contingency plan*.(4)

2 c) Describe four advantages of using an *agile* development processes in comparison with the *classical waterfall life-cycle*. There shall be at least one advantage from the customer's perspective and at least one advantage from the developers' perspective. (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. In a *UML state diagram* the *transitions* are denoted *event/action*, where *action* is the functions to execute during transition, and *event* is the underlying cause for the transition.
- B. An advantage of a *pipe-and-filter architecture style* is that the resulting code has no redundancy.
- C. The *Observer 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 A causes changes of the definition of B.



3 b) Describe a general advantage with the client-server architecture. Describe the principles of the three variants as taught in the course. (4)

3 c) Draw a *UML state chart* diagram of a telephone answering service with the following properties:

1. If the receiving subscriber does not answer after 4 signals, the answering service starts.
2. If the receiving phone is busy, the answering service starts immediately
3. The answering service plays a message and accepts voice input from the calling subscriber.
4. A new message can be recorded by the receiving subscriber, during which time the receiving phone is busy.
5. If the receiving subscriber answers the phone when the answering service is working, the service is cancelled, started recordings deleted, and the conversation can start as a normal phone call.

(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. In the software engineering terminology an *error* is a human mistake leading to a *fault* in the program, which if executed can cause a *failure*.
- B. *Usability testing* very often involves using a panel of test users performing representative tasks of the system.
- C. If you want to achieve *branch coverage testing* of a program you can be satisfied with a smaller number of test cases as you would need if you require *statement coverage testing*.
- D. A drawback of *Equivalence class testing* is that only valid input is tested.

4 b) Describe the different strategies for version handling: *Lock-Modify-Unlock* and *Centralized modify-merge*. (4)

4 c) To rent a car you must be 20 years or older, but not older than 75. Suppose we want to test software determining if you can rent the car. Identify the *valid* and *invalid equivalence classes*. Create a *test table* for *Equivalence class testing* with the minimum number of test cases. (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 *resolve* as many defects as there is time for.
- B. The *inspector* of an inspection is responsible for making sure that the *exit 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. An *audit* is performed by an independent, 3rd party organisation to evaluate, for instance, conformance to standards.

5 b) Explain how you calculate an approximation formula for *reliability* from time-based or interval-based failure measurements. Don't forget to motivate why it makes sense to use the approximation. (4)

5 c) Describe the following process areas from *CMMI*: *Project Planning*, *Organizational Process Focus*, *Requirements Management*, and *Requirements Development*. (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 design the system. You may add assumptions and constraints as long as they are explained.

- a) Name two quality factors that you think are important. Describe how you can determine whether a requirement on the quality factors is fulfilled or not. (5)
- b) Draw a diagram showing an execution view of the architecture of the system. Don't forget to explain the meaning with the parts of the architecture in text. (10)
- c) Describe how your architecture can help the design team to reach high quality goals according to the two quality factors in a) above. Are there any risks with the chosen architecture that can affect the quality factors? Good motivations are needed (10)

7. You are a newly hired R&D manager for a small software development company that sells embedded software for the retail market. Your main products are software that are placed in personal digital assistants (PDA:s) (for scanning inventories) and point-of-sales (POS) systems (the main system that the shop assistants are using when helping a customer). Currently, the company is using a development process that is similar to the classic waterfall model. The company do not have any tool support in their development except standard text editors, compilers, and a file server. There are 10 employees in the company:

- 1 CEO
- 2 Sales persons
- 1 Marketing manager
- 1 Project manager
- 2 testers
- 3 developers

Your task is to write down an e-mail to the CEO of the company, where you argue for that the company should change their way of working. Assume that the CEO is

mostly familiar with traditional project management, and that you suggest them to use a variant of Scrum, together with some useful agile practices. Make sure that you explain each concept carefully so that she understands why this is useful/not useful and how it might be implemented in the organization. Below, you will find a list of concepts and topics for which you are supposed to argue either for or against.

- Sprint
- Time-boxing
- GANTT-chart
- Waterfall model
- Self-organization
- Product Backlog
- Continuous Integration
- Automated Testing
- Scrum Roles
- Software Review

Please note that your email should be an arguing and convincing e-mail, not just a list of statements.. (20)