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 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 2013. 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 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!

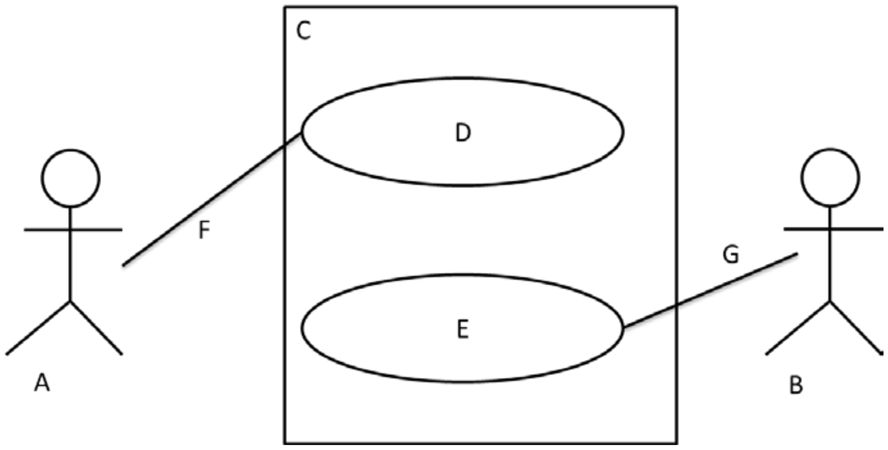
Kristian

Problems

Part 1: Fundamental

Area 1: Requirements

1 a) Consider the following generic use-case diagram:



Which of the following statements are true? Answer with the statement letter only, no motivation is needed. (2)

A. The model elements A and B are called *user agents*, acting on the behalf of a real user.
B. The model element C is called *subject*, showing which use-cases that will be implemented in the system.
C. The model elements D and E depict the *use-cases*. Their names appear in the ovals.
D. The model elements G and F contain *multiplicity* in the ends which limits how many times a use-case can be initiated.

1 b) *Scenario:* You are asked to specify a system for monitoring the ventilation in the C-house. Input sensors are measuring the temperature and humidity in the individual rooms and outside the C-house. The system can be programmed with different set-values of temperature and humidity per room. The control algorithms should save energy usage under the constraint of time to reach the set-values. The system can also be programmed for different modes depending on: which hours the room is booked a certain day and if the day is a working-day. There is also a surveillance and alarm system for monitoring the indoor climate. The main operator interface runs on a central computer, but many parameters can be accessed in mobile apps.

Task: Now, write down two *functional* requirements and two *non-functional* requirements of the system. It is important that the requirements can be verified once the system is implemented. (4)

1 c) Describe the following concepts: *Unambiguous* requirements, *consistent* requirements, *traceable* requirements, and a *complete* set of requirements. You may use an example. (4)
Hint: by "describe" we mean that a student like you but who has not taken the course shall be able to understand what you wrote.

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 risk "We will develop the current customer's requirements in a non-optimal order" is both a *project specific* and a *direct* risk.
B. The *risk magnitude indicator* is calculated by multiplying the probability and the impact of a risk.
C. By defining a *contingency plan* we will be able to lower the probability of a risk occurring.
D. It is good to manage many risks since many things might go wrong. Many risks can be 30-100 risks for a 10-person team.

2b) Describe the following practices of eXtreme Programming (XP): *Pair programming, Test-first programming, Refactoring,* and *Continuous integration.* (4)

2c) Describe the process of effort estimation using the *Delphi technique*. (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. The purpose of *design patterns* is to describe new and creative solutions to programming problems.
B. The *Façade* design pattern is useful when you want to provide a simple interface to a complex set of subsystems.
C. The *Observer* design pattern is useful when you want to create only a single instance of a class.
D. The *Strategy* design pattern reduces the number of conditional statements.

3 b) Define the concept of *cohesion* in the context of software architecture. Shall we strive for high or low cohesion in software design? Don't forget to motivate you answer. (4)

3 c) Draw a UML class diagram of the following (partial) rules for assigning teachers to thesis students. Use at least 2 generalisation associations:

- A thesis student must be assigned to an examiner for the work
- An examiner shall be employed as a teacher
- Teachers can be employed as lecturers or professors
- A master thesis student must also be assigned to a tutor
- Any employee, including assistants, can take the role as tutor
- A bachelor thesis student can have a tutor, but it is not required

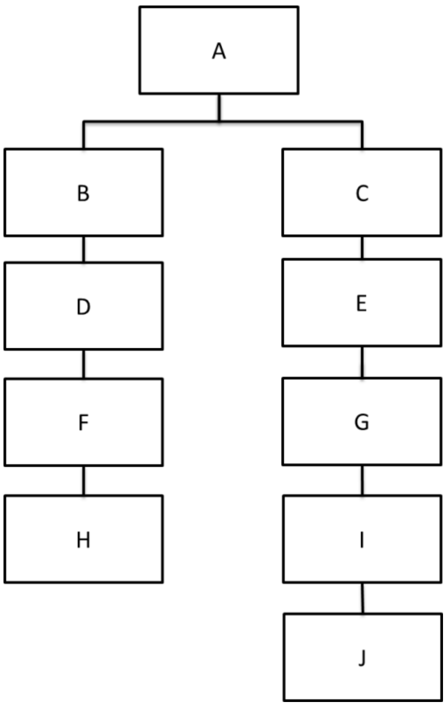
(4)

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)

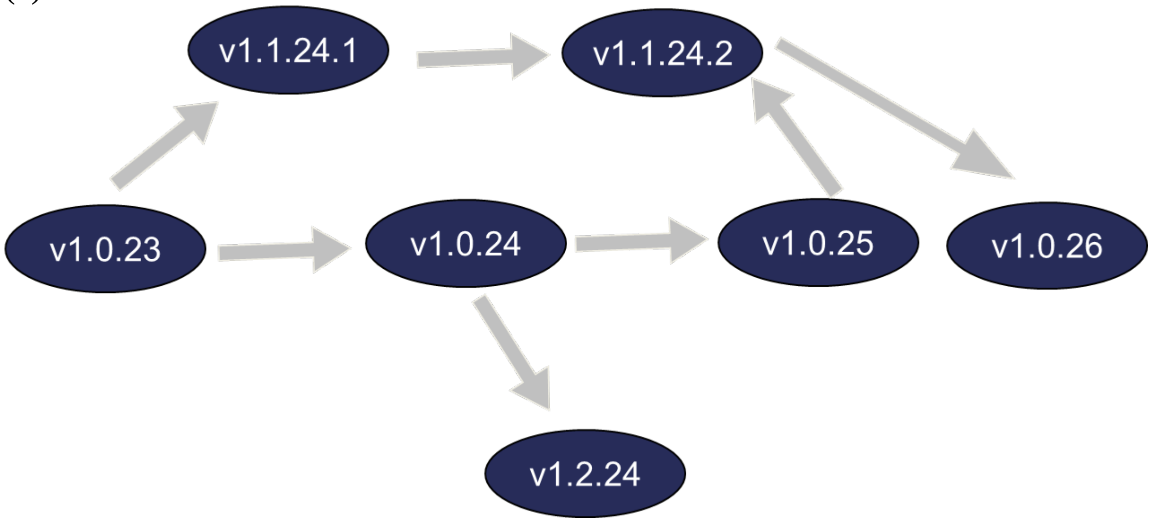
A. When performing *Equivalence class testing,* you try to create test cases covering as many invalid equivalence classes as possible in each test case.
B. To make *Boundary value test cases* you need access to the source code of the system.
C. It is possible to write a program that requires the same number of test cases for both *branch coverage* and *full path coverage*.
D. *Exhaustive testing* means that you test all possible input values to a program.

4 b) *Scenario:* Assume you have the following functional decomposition tree of a software system.



Task: Your task is to describe, perhaps with a diagram, the order of the different integration test sessions. Make one description for a top-down strategy and one description for the bottom-up strategy. Which strategy is most suited in this example? Don't forget to motivate your answer. (4)

4c) Describe the concepts *Trunk*, *Tag*, *Branch*, and *Merge* with help of the following picture of a history tree:
(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. In the *staged* representation of CMMI a *Maturity level* is associated with a set of *Process areas*.
B. To determine if a *Process area* in CMMI is satisfied, you check the fulfilment of *Specific* and *Generic* goals, associated with the process area.
C. The majority of the *Process areas* of CMMI are associated with level *5: Optimizing*.
D. The *Process areas* of CMMI are all independent of each other.

5 b) Shortly describe four types of data that are often collected from an inspection. (4)

5 c) Describe one metric that can be used to measure *usability* and one metric that can be used to measure *reliability*. (4)

Part 2: Advanced

6. Assign the following roles to the team members in the table. The table shows each person's competence ranking on a scale 0-10, where 10 is the highest competence. Motivate each assignment with 1-2 sentences:

- Product manager
- Configuration manager
- Analyst
- Architect
- Test leader

Skill/Person	**Nisse**	**Stina**	**Yuxiao**	**Nahid**	**Mary**
Analytical	5	8	10	6	8
Social	9	2	6	10	3
Communication	7	7	8	7	6
Technical	4	8	2	9	7
Sense of good order	2	6	6	6	7
Leadership	7	7	4	7	5
Sense for business	8	2	4	8	6

(10)

7. Write a description of the agile development framework SCRUM. Describe at least 8 different concepts, whereof at least two roles, two meetings, and two artefacts. Make sure that the descriptions of the concepts are connected somehow to at least one other concept. Reflect on what challenge(s) of software development that SCRUM tries to attain. Try to describe at least two challenges.
(20)

8. *Scenario:* You are in charge of designing a “continuous examination” system of a future university. The idea is that students shall be able to test their knowledge in different *subjects*, smaller than a course, whenever they feel prepared.

The student enters a small office with an interactive work-station using different interaction media. By using biometrics and other sensors, the system makes sure that there is only one person in the room and that the identity of the student is known. The student then logs in and navigates to the subject, for instance, Requirements Engineering of the TDDC93 course. The student interacts with the system for a specified time period and answers questions about theoretical concepts and solves sample practical problems. The tasks are sampled from the examiner’s data base and most of the examination is graded automatically with advanced classification software. Some parts are fed to teachers for grading in a structured way that makes the grading fast. The results from subject examination are accumulated and when a course is complete, the result is sent to the future LADOK for inspection and registration. The number of tasks in the database is limited and the students have a limited number of trials, since there are requirements on variation amongst the examination tasks.

Tasks:

a) Make a list of important roles of the system, such as clients, servers, databases, and users. Use this list to create a *UML sequence diagram* covering most of the system functions described in the scenario. At least five roles and ten messages are required. To get full credits you shall also use some kind of *fragment*. It is allowed to create a solution with several diagrams. (10)

b) A system like this requires much complex software of high quality. The system is developed gradually over several iterations to minimize the risks. Define the goal for the *first* iteration in terms of what parts that are most important to develop and test. Don’t forget to motivate the answer. (5)

c) Identify a sub-system and create a table of 5 test-cases for that sub-system. The test-cases are intended for use in test-driven design. (5)