

Written exam for Software Engineering Theory

Course codes TDDC88, TDDC93, 725G64

Note: When I visit the exam, I will take a slow walk among all students, so you don't need to sit with your hand raised. Just call for my attention when I pass your desk.

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 2014. 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 requirements of an app for an inventory software:

I: The app shall support hand-written input.

II: When the user has finished drawing a character, suggested interpretations shall be shown to the user in maximally 0.2 seconds.

III: The movement of the user's pointing device shall be displayed as a curve with a thickness of 6 pt.

IV: The movement of the user's pointing device shall be displayed as a curve with a thickness of 1 mm.

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

- A. The requirements III and IV are *consistent*.
- B. The requirement II is a *functional requirement*.
- C. The requirement I is a general property of the app called a *feature*
- D. The requirement III is a *functional requirement*.

1 b) *Scenario*: An inventory system of a super market consists of a central server where the inventory is always kept up to date. At arrival, the goods are automatically registered by a bar code scanner, and the truck driver gets an automatic order of where to store the goods. Items of goods are transported from the store to the shop by shop attendants. They use a mobile device to either: i) manually enter the number of items they place in the shop, or ii) scanning the bar code for each item placed in the shop. For fresh food, the weight of food in the shop is recorded. The cashier scans the bar codes of items being sold and enters the weight of fresh food. Attendants regularly makes inventory of remaining items in the shop. If there is a large difference between actual and expected number of items in the shop, the chief attendant is notified.

Task: Create a *use-case diagram* for a new inventory system consisting of two different *actors* and two different *use-cases*. Don't forget the use case texts. Only logging in and logging out are basic functions, not to be considered as use-cases. (4)

1 c) Good requirements have many properties. Explain the meaning of requirements being: *inspected*, *unambiguous*, *testable*, and *feasible*. (4)

Hint: by "explain" we mean that a student like you who has not taken the course shall be able to understand what you wrote. (1-2 sentences per concept)

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. An advantage with the *waterfall model* is that it is simple and easy to understand.
- B. An advantage with an *iterative model* is that the development team gets “lessons learned” and can improve the process while the project is running.
- C. A drawback of the *waterfall model* is that it starts with the least important requirements.
- D. A drawback of an *iterative model* is that the workload is uneven, all testing is left for the last iteration.

2b) Explain the concept of a *burn-down chart*, for instance from SCRUM. Make an example and describe how it shall be interpreted. Also, explain shortly how it can be used by a development team. (4)

2c) Write down four *risks* of a project: one *general*, one *project-specific*, one *indirect*, and one *direct risk*. Briefly describe the project you are thinking of. (4)

Hint: If you are taking a project course right now, use that one. If you are not, use your “own project”: Finish my courses for this semester.

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 *Façade design pattern* shows a flexible way of working with *inheritance*.
- B. The *Observer design pattern* can be used in an auction system to allow buyers to keep track of the currently highest bid.
- C. The *Strategy design pattern* is usable when you have a single intelligent algorithm that you want to use for many different purposes.
- D. The *Façade design pattern* can be used to implement a *layered architecture*.

3 b) Describe the concepts *coupling* and *cohesion*. Do we want high or low coupling amongst the components in an architecture? (4)

3 c) Draw a *UML state chart* of an item in the supermarket, described in the scenario in problem 1 b) above. Besides the information in the scenario, an item can be:

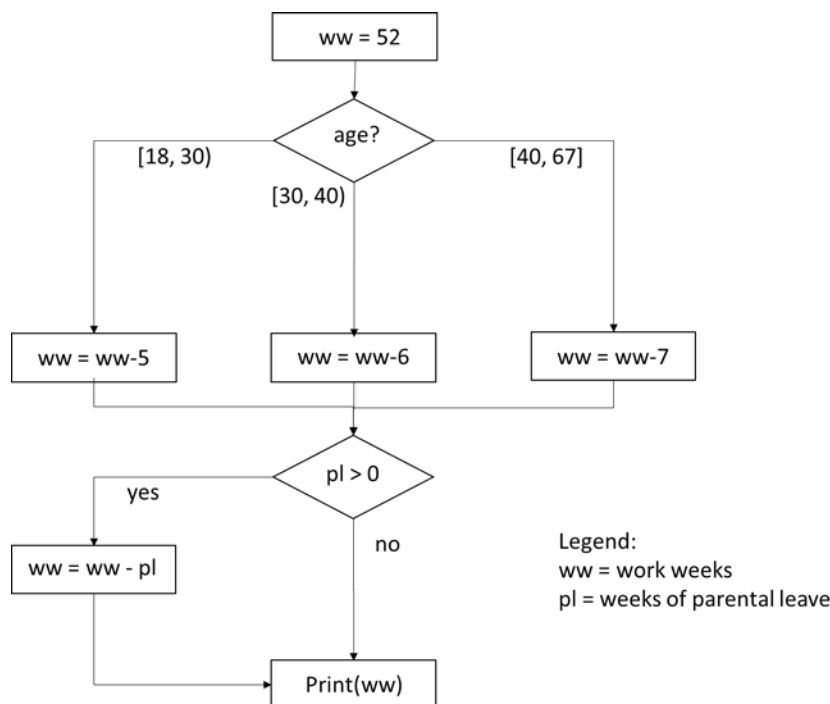
1. Stolen, never found again, and thus erased from the inventory.
2. Stolen, reclaimed by the police and put back to its proper place in the store.
3. Destroyed if the customer or shop attendant finds a defect.
4. Destroyed if it has been too many days in the shop.

Don't forget to write clear and logical information on the *transitions*.

You may make more assumptions of the system than those given in the problem description, but in that case you shall explain your assumptions in the solution. (4)

Area 4: Testing and SCM

4 a) Consider the following *flow graph* and test table¹



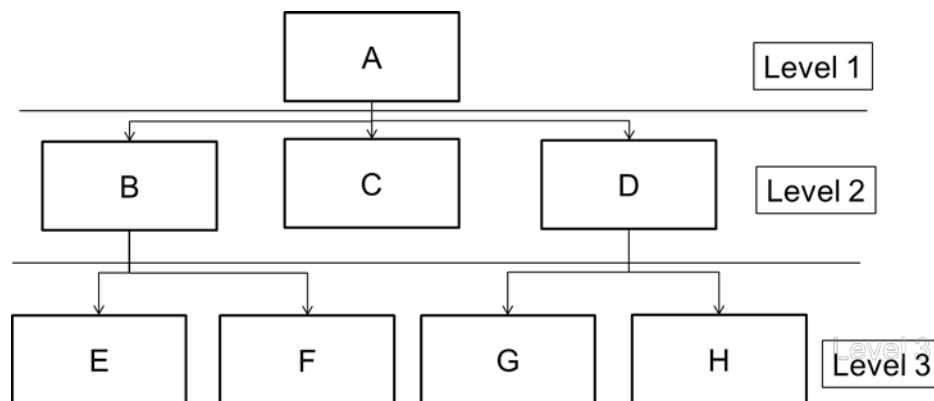
Id	age	pl	ww (expected)
1	25	0	47
2	35	10	36
3	35	0	46
4	50	0	45

¹ [means included; (means not included. The inequality $2 \leq x < 6$, is written [2, 6)

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

- A. The *test cases* 1, 2, 3, and 4 are **not** sufficient for *full path coverage* testing.
- B. The *test cases* 1, 2, 3, and 4 are the minimum number of *test cases* for *statement coverage*.
- C. The *test cases* 1, 2, and 4 give *branch coverage*.
- D. Given that the flow graph follows the specification exactly, there would be 3 *equivalence classes* if we follow the rules from the lecture.

4 b) Consider the following functional decomposition tree of a system:



- a) How many *stubs* do you need if you perform *top-down integration testing*?
- b) How many *drivers* would you need if you perform *bottom-up integration testing*?
- c) How many *drivers* would you need if you perform *big-bang integration testing*?
- d) If it costs x times more hours to code a *driver* compared to a *stub*, when is it more advantageous to use *bottom-up integration testing* compared to *top-down integration testing*?

Don't forget to motivate all your answers. (4)

4c) Explain the input, output, and functions being processed of a *build and integration server* as it is used in *continuous integration*. (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 goal of an *inspection meeting* is to resolve and correct as many *defects* as possible, even if a few might not get documented in the inspection record.
- B. The reason for collecting data about an *inspection* is twofold: to improve the development process, and to improve the inspection process itself.
- C. A *walk-through* is less formal than an *inspection* and is effective in finding defects caused by *errors of omission*.
- D. The *inspection leader* doesn't check for defects himself/herself.

5 b) Calculate the *cyclomatic complexity* of the flow graph used for illustrating problem **4 a)**. Will the cyclomatic complexity change if you add a statement, for instance, to pay a bonus if $p1 > 0$? Don't forget to motivate the answer. (4)

5 c) Suppose that you work in a company at *CMMI level 1* that has high requirements on *usability* of the products. Select the two most important *process areas* of *CMMI level 2* or *3*, that you would like to focus on first. Briefly describe the process areas and motivate how they can contribute to high *usability*. (4)

Part 2: Advanced

6. *Scenario*: Let us return to the description of the inventory system in problem 1 b). In that text you could read about some different end-user categories (truck driver, attendant, and cashier). Including these, write down a list of 5 different *stakeholders* of a project developing a new inventory system. Thus, you have to define at least 2 more *stakeholders*.

For each *stakeholder*, write down one of their most prioritized *software quality factor*, together with a motivation of about 2 sentences for why this factor is important. Your solution must deal with at least three different quality factors to demonstrate that you know more than your favorite *quality factor*.

Example of accepted pattern of solution:

Stakeholder1	Quality factor1	Motivation1
Stakeholder2	Quality factor1	Motivation2
Stakeholder3	Quality factor2	Motivation3
Stakeholder4	Quality factor2	Motivation4
Stakeholder5	Quality factor3	Motivation5

(In the example stakeholders and motivations are unique, whereas Quality factors 1 and 2 occur twice. Still the solution contains 3 different quality factors, and is thus accepted)

(end example)

(10)

7. Two developers sit on a train and listen to a conversation of two *SCRUM-masters*. The train is quite noisy so they do not get all the details, but they decide to try to use SCRUM in their project anyway. However they do not agree about what they heard about the length of the *sprint*, the size of the *team*, the length of the *daily SCRUM meeting*, and who was doing the *estimates*. So, they send an e-mail to you with the following questions:

What would be the consequences, good or bad, if we:

- ... set the length of a *sprint* to 1 week compared to if we set it to 4 weeks?
- ... have *daily SCRUM* meetings of 15 minutes compared to if we *time-boxed* the meetings to 60 minutes?
- ... have a *team* size of 4 people compared to a *team* of 14 people?
- ... the *estimations* are done by the *SCRUM master* compared to if they are made by the *team*?

Help them by answering the questions.

The motivations for the consequences are important and are the base for grading.

(20)

8. Scenario: The trains in Sweden must be on time. Therefore you have started developing a passenger delay minimization system. The idea is to keep track of the passengers with tickets in their mobile phones. When a train is approaching the station, passengers are reminded to go to the platform. Passengers get reminders at different times depending on i) their distance to the platform and ii) a prognosis of queues to stairs and elevators; we want a constant flow of people moving.

At the platform the passengers get information about where to wait for boarding the train. The system knows which doors of the train that is best for each passenger. Passengers can either get a map on the screen or be guided with a clicking sound, where the frequency becomes higher the closer they are to the correct boarding position.

If passengers send a “need assistance” message to the server, the train crew and the station hosts gets a list on their mobile units of people that need boarding assistance. The people needing assistance are prioritized depending on how far they are from their boarding position. A station host or a crew member can select a person to help, then the other personnel do not need to bother about this person. When done, the helper checks the passenger on the mobile device, and a confirmation is sent to the chief crew member.

For safety reasons, you don’t want people to run around when the train is in motion at the platform. So, you implement a safety software according to the following specification: 5 seconds before the train reaches the first end of the platform, all information about boarding positions are erased from passengers’ phones. The information appears again when the train has been standing still for 2 seconds.

Task:

- a) Identify about 5 *hardware nodes* that you need in order to *deploy* software to your system. (5)
- b) Draw a *UML sequence diagram* for the *messages* between the hardware nodes. Only passengers with a pre-booked assistance need are allowed to send “need assistance” messages. Use at least one *fragment*. (5)
- c) Write *test-cases* for the software controlling the safety software. Identify *input* and *output variables*. The test cases shall fulfill the *boundary value testing* as is defined in this course. (10)