Linköpings Universitet
IDA
Kristian Sandahl

2018-01-03

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 2017. 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) Take a look at these two *user story* cards:

#1 As a teacher I want to record student lab results so that I can make a correct result file at the end of the course.

Priority: 1
Estimate: 40

#2 As a teacher I want to make sure that only allowed collaboration between labgroups are used so that I don't need to report anyone for cheating.

Priorty: 2
Estimate: 800

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

A. *User story* #1 is twice as important as *user story* #2.
B. Both user stories follow the pattern "As a (role) I want (something) so that (benefit)"
C. The *estimate* shows how much the customer is willing to pay for the story.
D. *User story #2* is too general to be used in an *agile* development project.

1 b) *Scenario:* You are developing a home page for consumer product test magazine that provides readers with results of professionally performed tests with recommendations.
Anyone can see headlines, read summaries of the tests, and buy an article with the full test with a credit card. The purchased article can be accessed for seven days. Subscribers have full access to all test articles during the subscription period and can also make comments to the tests sharing their own experience of the product with other subscribers. The editor publishes the tests and writes the summaries and explaining information about the test. The editor can remove subscribers' comments and archive outdated tests.

Task: Your task is to draw a UML *use-case diagram* of the site comprising at least two different *use-cases* and two different *actors*. Don't forget the use-case texts (also known as descriptions). (4)

1 c) Describe the following concepts in the context of software requirements engineering: *unambiguous requirements, prototyping, design constraints,* and *requirements validation.* About 1-2 sentences per concept is probably enough. (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)

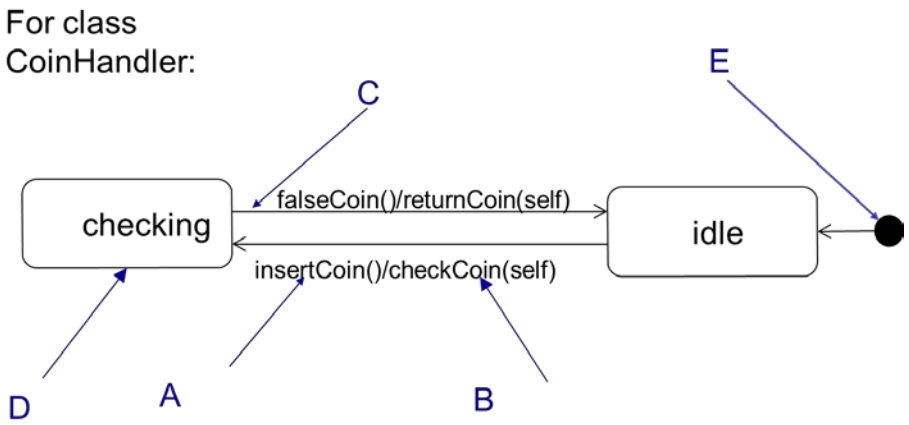
A. The *waterfall model* can be good for fixed-price projects.
B. A problem with the *waterfall model* is that it is hard to understand.
C. A prerequisite to use *iterative development* is that the requirements can be divided in smaller parts.
D. A problem with the *iterative model* is that work can not be done in parallel.

2b) Describe input, activity, and output of the four *risk management* process steps. (4)

2c) Describe the *SCRUM* meetings: *daily SCRUM, sprint planning meeting, sprint review meeting*, and the *sprint retrospective meeting*. About 1-2 sentences per concept is probably enough. (4)

Area 3: Design and Architecture

3 a) Look at the UML *state machine diagram* below:



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

A. A points at a *triggering event* that causes the *state* to change.
B. B points to a *sub-triggering event* that also has to be fulfilled to cause a *state* change.
C. C points to a *message* arrow that shows that parameters are passed between the *states*.
D. D points to a *state*, whereas E points to a *pseudo state.*

3 b) *Modularization* is a wanted property of many *software architectures*. Describe two advantages of *modularization*. About 3-4 sentences per advantage will probably do. (4)

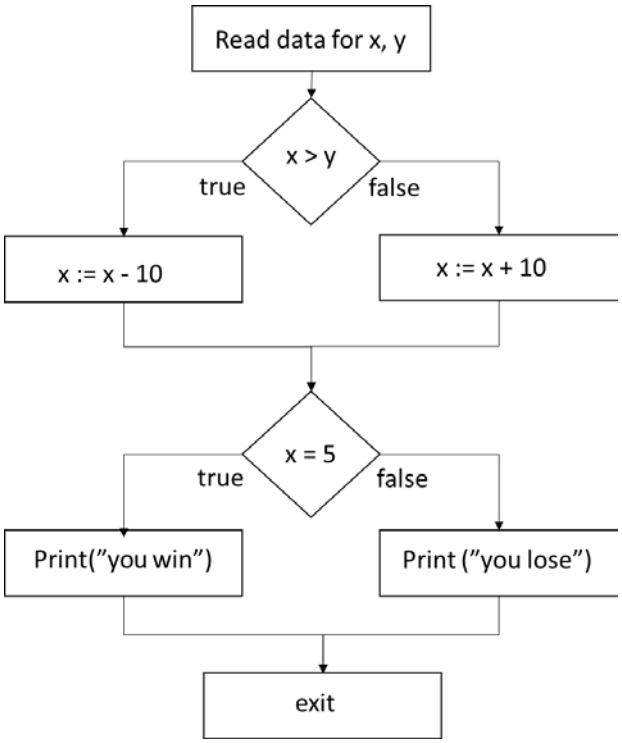
3 c) Describe an *object-oriented design pattern* with a UML *Class diagram*. Describe one *applicability* and one *consequence* of using the design pattern. You may use an example. (4)

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 a *decentralized modify-merge* version handling system when you *clone* a public repository you fetch and merge recent changes with your local repository.
B. In a *decentralized modify-merge* version handling system you create *tags* to mark a point in the history tree as important.
C. *Continuous integration* implies that you build and test the system often to keep the *code base* stable.
D. In *continuous integration* it is important that you run exactly the same *test cases* for each integration so you can show *reliability growth*.

4 b) Look at the following *flow chart* of a program



Create a *test table* of *test cases* that ensures *full path* coverage. (4)

4 c) Write down two advantages and two potential problems with a *bottom-up integration testing strategy*. (4)

Area 5: Software Quality

5 a) You have the following metrics for two programs, A and B:

Metric	Program A	Program B
Lines of Code	20 000	10 000
Cyclomatic complexity	10	30
Mean Time To Repair	2 hours	4 hours
Percentage of commands not invoked by end-users	5	20
Time to implement	2 person months	2 person months

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

A. Program A needs many more *test-cases* than program B to ensure *full path coverage*.
B. Program A has both better *availability* and *maintainability* than program B.
C. It is reasonable to believe that program A is more *relevant* for its users compared to program B.
D. It is reasonable to believe that the programmers for program A were less productive[1] than the programmers for program B.

5 b) Describe how a *CMMI Process Area* is structured in the standard document. What is the criterion to determine of whether a *Process Area* is fulfilled or not? (4)

5 c) What are the responsibilities of the *inspection leader* (also known as moderator) in the four steps of an *inspection* process? (4)

[1] A measure of the efficiency of a person, machine, factory, system, etc., when converting inputs into useful outputs.
Productivity is computed by dividing average output per period by the total costs incurred or resources (capital, energy, material, personnel) consumed in that period. Productivity is a critical determinant of cost efficiency. (http://www.businessdictionary.com/)

Part 2: Advanced

6. *Scenario:* Vaccination against the disease TBE (Tick Borne Encephalitis) needs to be carefully planned so the patient gets a good protection. The following table apply[2]:

Dose number	Patients below 50	Patients 50 and above
1		
2	1-3 months after dose 1	1-3 months after dose 1
3	5-12 months after dose 2	5-12 months after dose 2
4	3 years after dose 3	3 years after dose 3
5 and more	At least every 5th year	At least very 3rd year

You have developed an app that keeps track of the patient's schedule. When the patient takes a dose the nurse marks the date. The app sends a reminder to the patient when there is less than one month before the deadline of the next dose. The patient can also ask the app if the next dose can be taken at a specific date. The app knows the birthdate of the patient and the present date.

Task: Identify all *input* and *output variables*. Identify 5 valid *equivalence classes*. Provide a *test-case table* with one *test-case* per identified *equivalence class*.
(10)

7. Describe four different *metrics* that can be used to assess the *usability* of an interactive software product, such as a mobile/computer game. The metrics shall be obtained in different ways:

- One metric shall be possible to obtain with a test panel running a *final release* of the software and then answering a questionnaire.
- One metric shall be possible to obtain by making *observations* of a test panel running a *final release* of the software.
- One metric shall be possible to obtain with a test panel during different stages of *prototyping*, including *low-fi prototypes*.
- One metric shall be possible to obtain without involvement of user representatives.

For each of the metrics answer the following questions:

a) What do you measure? (Called "Description" in the metrics slides.)
b) What procedure do you need to preform to get the data?
c) What resources do you need to collect the data?
d) How do you calculate the numerical number(s)?
e) How does the metric relate to usability? Use arguments such as: "A high number of <my metric> indicates high usability since …"

(20)

[2] For a common type of substance, no responsibly taken.

8. *Scenario:* You are building a system of robots to carry fragile goods in an unknown indoor environment. All robots have a communication device. There are two kinds of robots: Transport robots and scouts. Transport robots carry goods and have a central processing unit that calculates the shortest safe way. Scouts come in two types: those that fly and those that moves on the ground. Scouts can be configured with one of two skill sets. The first skill set involves detecting the borders of the room: walls, ceiling and conditions of the floor. The second skill set involves finding hindering objects in the room.

Task: Create a UML *class diagram* modelling the equipment described above. Each physical equipment is a *class* and you use at least one *generalization* and one *composition* association. (10)

9. *Scenario*: The same is in problem **8.** above. You implemented a system of three robots:

- The first robot is a scout and runs on the ground. It is small and fast and have sensors that can detect distances between walls and ceiling. It can also detect the conditions of the floor.
- The second robot is an airborne scout. It is small and fast and have sensors that can detect objects in the way of transportation.
- The third robot is a transport that runs on the ground. It is large and slow and takes the goods. It also contains a computer that sends orders to the other two robots, and predict the shortest safe way to go.

The first robot is ahead and checks if it is at all possible for the third robot to enter. If not, a re-plan message is sent to the third robot, which then backtracks and sends the two to scouts to find a new path. If there is enough space the second robot receives an order to scan for hindering objects. Based on the results the third robot calculates a safe way and orders the first robot to advance.

Task: Model this communication loop in a UML *Sequence diagram*. Use at least one *fragment*. If you make assumptions about the system, write them down. (10)