Linköpings Universitet 2015-08-27
IDA
Kristian Sandahl

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!

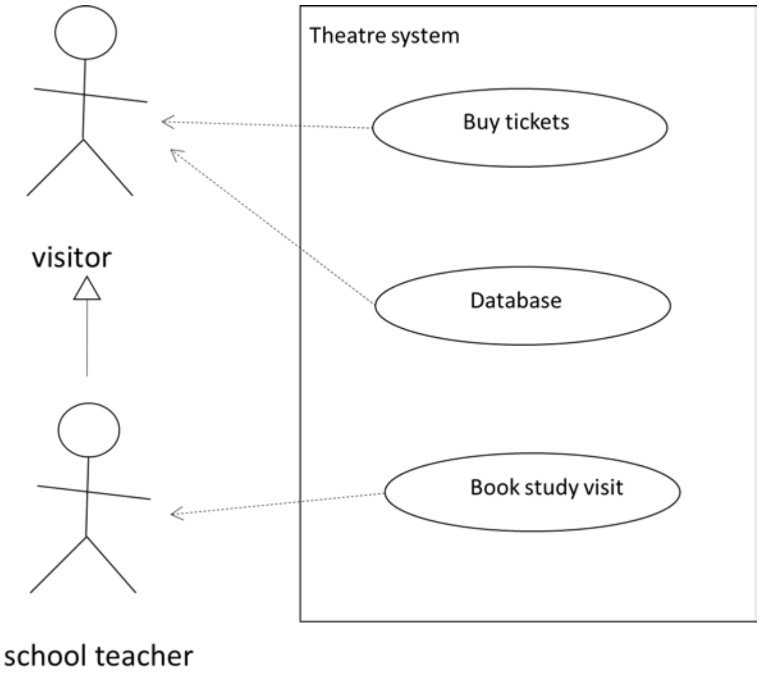
Kristian

Problems

Part 1: Fundamental

Area 1: Requirements

1 a) Consider the following *UML use-case diagram* for a theatre system:



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

A. The arrows from use-case ovals to actors are correct.
B. The arrow from actor school teacher to actor visitor is correct.
C. All use-case names are correct verb phrases.
D. The actor school teacher can buy tickets.

1 b) Write 2 *functional* and 2 *non-functional* requirements for a web-shop selling mobile phones and mobile phone subscriptions. (4)

1 c) Explain what the following concepts are: *Consistent requirements, Use-case description, Requirements validation,* and *Requirements elicitation*. (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. A *risk* of a project is something that can eliminate the full success of the project.
B. When you *mitigate a risk* you are lowering the probability that it occurs.
C. It is good if you can monitor many (30-100) *indirect risks* in the project.
D. The *risk magnitude indicator* is the product of the risk probability and the likelihood of the risk.

2b) Describe the idea behind *iterative development*. Also describe 2 advantages and 1 potential problem with using an iterative development method. (4)

2c) Describe 4 similarities and/or differences between *SCRUM* and *Kanban*. (4)

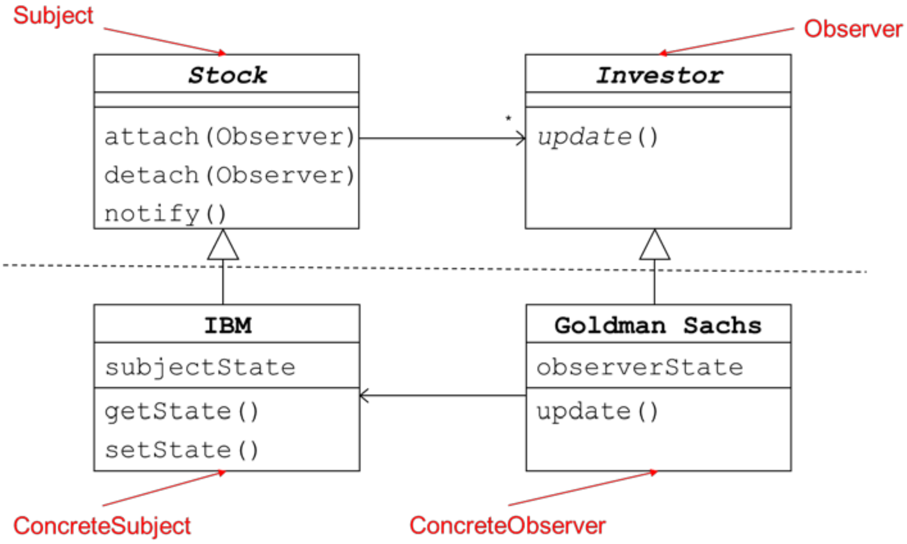
Area 3: Design and Architecture

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

A. In a *service-oriented architecture* there is lot of data communication between the different services.
B. An advantage with a *pipe-and-filter architecture* is that it eliminates redundant code.
C. A *layered architecture* can be used to provide a modern interface to old, but working, software.
D. A *client-server architecture* with a thin client is typical of a mobile app, where the clients' computational power usually is not so great.

3 b) Draw a *UML Sequence diagram* of the communication between the customer and an automatic teller machine for the successful withdrawal of money from the customer's bank account. There shall be at least one *fragment* for alternate branches. Hint: You don't need to involve the bank's database, just focus on the two nodes. (4)

3 c) Consider the following class diagram of the *Observer design pattern*.



Describe the function of methods `attach(Observer), detach(Observer), getState()`, and `setState()` (4)

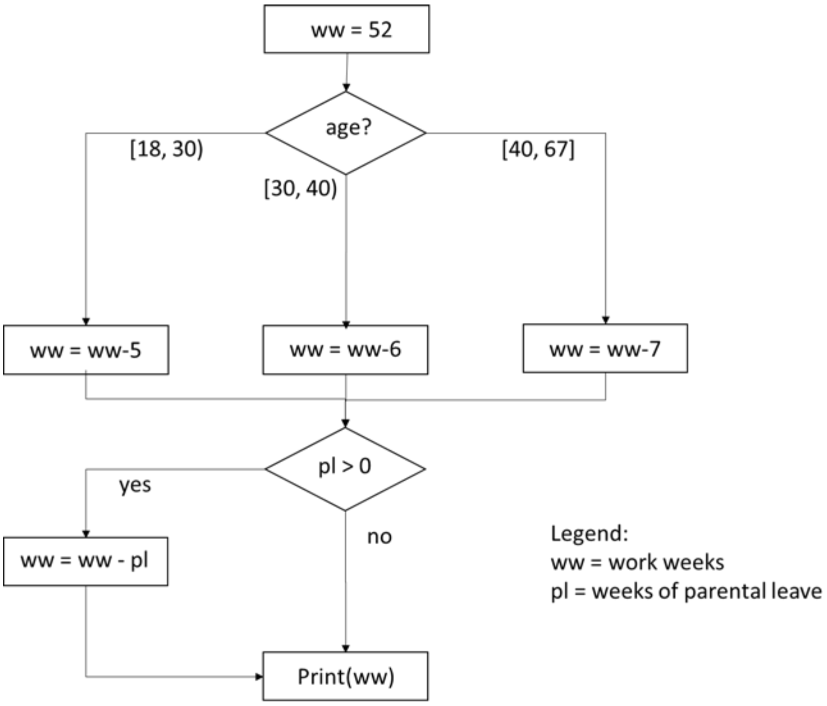
Area 4: Testing and SCM

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

- **A.** With a *lock-modify-unlock* revision control system a user can lock a file to prevent others from making modifications of the file.
- **B.** In a decentralized *modify-merge* revision control system users clone their *local repositories* to the *public repository* when modifications are done.
- **C.** The `fetch` command in *Git* performs both a `pull` and a `merge` operation.
- **D.** In a *centralized modify-merge* revision control system several users can *check out* a file and work in parallel.

4 b) Describe the *acceptance test* approaches: *Pilot testing* and *Parallel testing.* (4)

4c) Consider the following *flow-graph* of a system that calculates the working weeks of a year based on age of the worker and weeks of parental leave[1]:



Make a table of test-cases for *equivalence testing*. If you make assumptions about the system, write them down. (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 reached if all *process areas* belonging to that level and lower levels are *satisfied*.
B. Your organization is at CMMI level 1 in the staged representation given that basic documentation, such as requirements and project plans, are at a satisfying level.
C. If, and only if, you can map the *specific goals* of a process area to your way of working, you have *satisfied* that *process area*.
D. For CMMI level 3 in the staged representation it is essential that you have your own standard processes that you tailor to specific projects.

[1] [means included; (means not included. The inequality 2 <= x < 6, is written [2, 6)

5 b) Describe two different *metrics* that can be used to assess the *usability* of a software system. Don’t forget to describe how you obtain the metrics and why a high or low value indicates good usability. (4)

5 c) Explain what is done in the four phases of a software *Inspection process: Plan and overview, Individual checking, Inspection meeting,* and *Edit and follow-up.* (4)

Part 2: Advanced

6. Make an outline of the headings of a *project plan*. For at least 4 of the headlines.

a) give examples of information that belongs to the sections of the headline, and
b) indicate if it is highly likely or not likely that the information will change during the project. Don’t forget to motivate your answer. (10)

7. In *agile methods* you often encounter the practices of writing *user stories* and using *test-driven development*. Describe these two practices and list expected benefits. Also list potential drawbacks that might occur when following the practices. (20)

8. *Scenario:* An issue tracking system, ITS, is a software managing lists of issues, as needed by an organization. Issue tracking systems are commonly used in an organization's customer support call center to create, update, and resolve reported customer issues. Alternatively, issues can also be reported by that organization's other employees. A support ticket shall include information for the issue encountered. An issue tracking system often also contains a knowledge base containing information on each customer, resolutions to common problems, and other such data.

A ticket element, within an issue tracking system, is a running report on a particular problem, its status, and other relevant data. They are commonly created in a help desk or call center environment and almost always have a unique reference number, which is used to allow the user or help staff to quickly locate, add to or communicate the status of the user's issue or request.

Tasks:

a) Draw a *UML use-case diagram* of an ITS with at least 2 use-cases and 2 actors. Don’t forget the use-case texts. (5)
b) Draw a *UML class diagram* of the concepts being handled by the system. Don’t forget to write name and *multiplicity* of the *associations*. (10)
c) Draw a *UML state diagram* for a ticket element. (5)

(if you make assumptions about the system, write them down)