

## 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 2015. 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)** Which of the following statements are true? Answer with the statement letter only, no motivation is needed. (2)

- A. The requirement “The system shall print the current balance of the customer’s account on the receipt.” is not a *testable* requirement even if we have access to the customer account database.
- B. The requirement “The system shall make a prognosis of the weather for the next 24-hour period.” is a *feasible* requirement.
- C. The requirement “The system shall make the user surprised by waiting a while to display the result.” is an *unambiguous* requirement.
- D. The requirement “The comments in the code shall be in Swedish.” is a *non-functional* requirement.

**1 b) Scenario:** You are developing an information system for a support service for mobile phones. The support service consists of different subsystems:

- A navigation system where the user answers a series of questions to pinpoint the fault, and a remedy. The content is edited by the support personnel
- A searchable database of Frequently Asked Questions (FAQ). The content is edited by the support personnel.
- A searchable user forum. Anyone can read contents. Persons with a free user-group membership can add questions and answers.
- A form for sending free text messages.

*Task:* Write at least two *use-cases* of the system with at least two different *actors*. Draw a *UML use-case diagram* of the *use-cases* and *actors* you created. (4)

**1 c)** Explain what the following concepts are: A set of *consistent* requirements, a set of *complete* requirements, a *formalized* requirement, and a *feature*. (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 *product backlog* in SCRUM is a list of all things that needs to be done within the project.
- B. The *burndown chart* in SCRUM is a visual measurement tool that shows the completed work during a certain time period.
- C. The *daily SCRUM* is a meeting with the *product owner* where priorities are discussed.
- D. SCRUM is a *systems development method* that is specialized for software development.

**2b)** Explain what a *critical path* and *slack (float) time* are. (4)

**2c)** Describe two advantages and two potential problems of the (*classical*) *waterfall* life cycle model. (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 no data communication between the different services.
- B. A *layered architecture* can be used to provide a modern interface to old, but working, software.
- C. An advantage with a *pipe-and-filter architecture* is that it eliminates redundant code.
- D. A *client-server architecture* with a fat (thick) client is typical of an app on a TV-box, where the clients' computational power usually is not so great.

**3 b)** Make a *UML sequence diagram* using at least two *fragments* of a web shop selling and booking cinema tickets where you pay with credit card. (4)

**3 c)** Already when you make the architecture of a system you can attain different quality factors. For instance, if performance is in focus you can design for parallel execution. This question is: How can you make a *software architecture* that is suitable for developing a system with high *usability*? Write a short motivation (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. An advantage of using a *top-down integration testing strategy* is that you do not need to implement *drivers*.
- B. A disadvantage of using a *big-bang integration testing strategy* is that you do need to implement *stubs* for all components.
- C. A disadvantage of using a *bottom-up integration testing strategy* is that you postpone the opportunity to discover performance problems of lower-level components.
- D. An advantage of using a *top-down integration testing strategy* is that you can discover problems with the overall design early in the process.

4 b) Draw a flow chart of a procedure where the minimum number of test cases for *branch coverage* is strictly smaller than the minimum number of test cases for *full path coverage*. Do not forget to motivate your answer. You may use an example. (4)

4 c) Define the concepts of software testing: *Error*, *Fault*, *Failure*, and *Oracle*.(4)

## Area 5: Software Quality

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

- A. For programs with a connected *flow graph*, that only contains *binary decisions*, the *cyclomatic complexity*,  $V(G)$ , grows linearly with the number of binary decisions.
- B. *Cyclomatic complexity* cannot be calculated for *object-oriented* software, such as Java programs.
- C. Low *coupling* between modules makes maintenance more easy since the number of code lines will be reduced.
- D. *Reliability* is the probability that the system will work as intended during a specified time interval

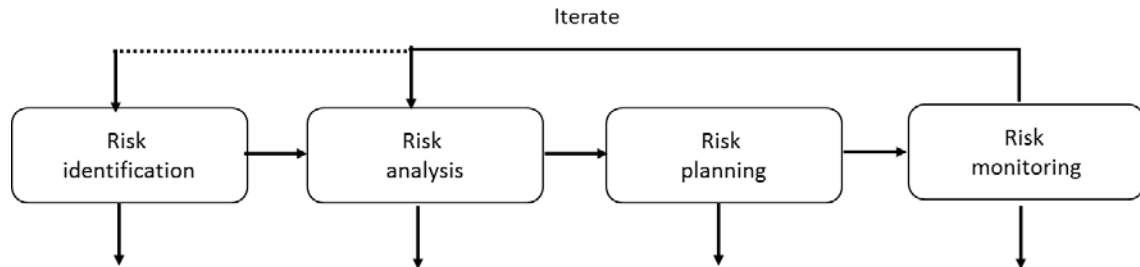
5 b) *Scenario*: You run a company which delivers information systems to large organizations. The end-users are normally very happy about your solutions, but the IT-technicians at the customers' data center complain: your software uses old technology, is complicated to install, and you are incapable of delivering all components at the same time, which causes delays. Next release has to focus on installation and operation.

*Task*: Identify a *CMMI process area* that can help you to improve on next release. Summarize the process area's purpose and introductory notes in 5-6 sentences, and motivate how this could help you. (4)

5 c) Describe the purpose and main steps of the software review methods: *inspection*, *management review*, (*structured*) *walk-through*, and *audit*. (4)

## Part 2: Advanced

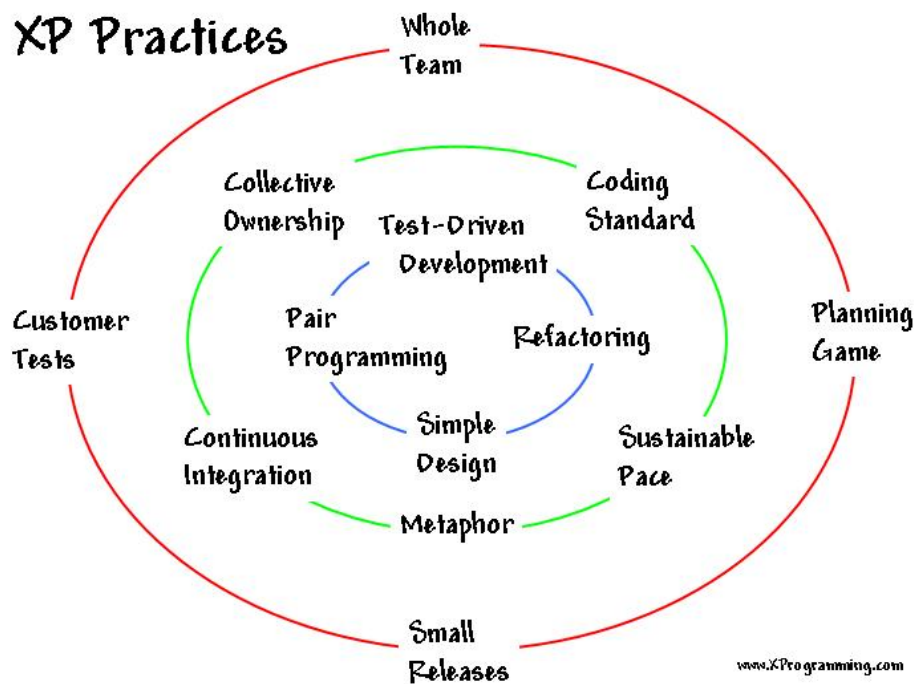
6. The figure below shows a part of the *risk management process*. For each of the steps, write a short description of the major activities and a description of the output. Also, give an example of a pitfall that you have to consider when working with risk management (a risk of risk management). (10)



7. The figure below depicts different *practices* of eXtreme Programming, which is an agile development method. Select five of the practices and for each practice describe:

- how the practice is performed.
- expected benefits for a special *stakeholder* (don't forget to describe the stakeholder you have chosen)
- how the practice challenges the traditional way of working
- a potential risk or precaution that must be considered when applying the practice in a student project.

(20)



**8. Scenario:** A popular attraction during summers are all opera festivals taking place in different places in Europe. Since the visitors are focused on the music they don't want to spend time planning practical details. You decided to make an online festival booking and travel agency systems with lot of automatic planning.

A travel comprises transport, housing and event tickets. Many of the users combine several travels into a tour, to visit more than one place without returning home. Users can travel individually, but are mostly part of a group. Often, groups like to join other groups at parts of their travel in super-groups. For instance, group A and B travel jointly to England, then group A goes to Germany while group B travels to Italy, and finally they join in Austria.

Your system shall be able to suggest tours, both to find events users may like but also to find opportunities for groups to travel together. Therefore, you have to keep track of the relations between groups in your system. Groups that are *close friends* want to share all arrangements of the tour. Groups that are *just friends* want to travel together, but can have different hotels. If two groups are *acquainted* to each other they still enjoy going to the same performance, but can live at different hotels and travel with different transportations. A festival can contain different performances.

*Task:* Make a UML *class diagram* of the concepts handled by the system. Don't forget multiplicity and names of association. Use both attributes and operation of the classes. (10)

**9. Scenario:** In the same context as in question 8) above: It can be tricky to match groups. One problem is that the German composer Richard Wagner (1813-1883) is still controversial. Your users are thus divided in three categories:

- I. People who would never go to a Wagner opera performance.
- II. People who are really interested to go to a Wagner opera performance if possible.
- III. People who might go to a Wagner opera performance, but only together with close friends. They value the friendship high, and can endure Wagner if necessary.

*Task:* You are supposed to write test cases for the function `WagnerFit`. The function takes two groups as input, and depending on their category and relation, the function returns true if the groups can travel together and false otherwise. Identify 4 equivalence classes and provide a test table for them. Also explain how you interpret the function of "can travel together" since different assumptions are possible. (10)