

Written exam for Software Engineering Theory

Course codes TDDC88, TDDC93, 725G64

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

If you get trouble with Wiseflow that you or the exam invigilator cannot solve in 10 minutes, call the examiner and continue with other problems.

Instructions to students, please read carefully

- **Explicitly forbidden aids:** Textbooks, machine-written pages, photocopied pages, pages of different format than A4, other electronic equipment than the computer for running Wiseflow. A silent keyboard, mouse and a webcam might be brought.
- Try to solve as many problems as possible.
- Motivate all solutions.
- Please, draw clearly.
- You may write solutions in either Swedish or English, but don't mix in the solution of the same problem.
- Please, note that the problems are not necessarily written in order of difficulty.
- TIP! Read 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 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 2021. 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 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 a maximum of 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 statements C and D were false, you would get +1 credit for the multiple-choice question.

Example 3: Assume you correctly wrote both statements A and B. If now statements A and B were true, and statements 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

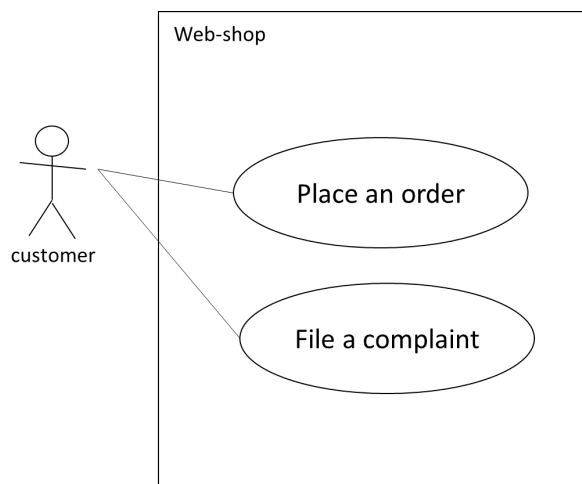
Problems

Part 1: Fundamental

Area 1: Requirements

1 a) Look at the UML use-case diagram below. Which of the following statements are true? Answer with the statement letter only, no motivation is needed. (2)

- A. The rectangle named Web-shop is a class that will contain all methods to realize the use-cases.
- B. From the diagram we know that the actor can initiate interaction with the system to fulfil the use case "Place an order".
- C. To determine whether the use case "Place an order" is fulfilled we need more information than is shown in the diagram.
- D. The lines between actor and use-cases have a non-sharing property, which means that a new actor, purchaseResponsible, could not share the "Place order" use-case with the customer actor



1 b) Explain the following concepts in the context of software requirements: *Traceability*, *stakeholder*, *functional requirement*, and *user story*. Hint: an explanation is typically 1-2 sentences. (4)

1 c) Write two verifiable *non-functional* requirements of a social media system, for instance Facebook. One of the requirements shall specify a *quality factor* and the other shall be a design constraint of the system. (4)

Area 2: Design and Architecture

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

- A. A *UML class diagram* governs the way objects can be instantiated.
- B. The *generalization* relation in a UML class diagram is used to describe inheritance.
- C. An *abstract class* in UML can only have abstract operations.
- D. By default, an *association* from class A to class B, means that instances of B visible from A are *ordered* and *nonunique*

2 b) Describe two expected benefits from designing and documenting a *software architecture*. Don't forget to motivate the answers. (4)

2 c) Draw a *UML state diagram* of the class Book in a library system. There shall be at least three *states*, that are not *pseudo states* (for example a start state or a stop state). The syntax must be correct and the semantics reasonable. (4)

Take a photo of the diagram and add that as an appendix to the answer. Let the name of the appendix start with the characters "2c".

Area 3: Testing and SCM

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

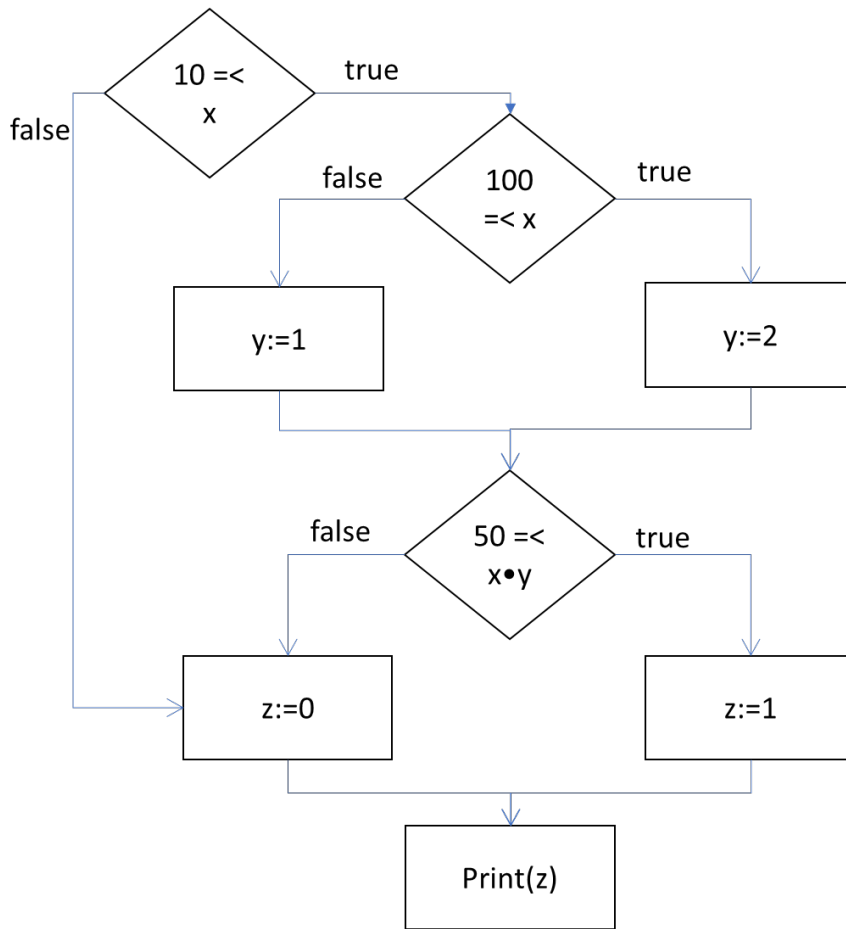
- A. The *centralized workflow* in revision control is good for large projects since the *master* (trunk) collects all commits in a structured way.
- B. The *Git workflow* in revision control is good for simultaneous development of several product versions as we can have many branches.
- C. An advantage with the *feature branches* workflow in revision control is that it is possible to use *pull requests*.
- D. An advantage of the *decentralized workflow* in revision control is that there is no need to provide write access to the *main repository* since developers use their own local and public copies of the repository.

3 b) Describe a benefit for each of the *integration testing* strategies: *Big-bang*, *Bottom-up*, *Top-Down*, and *Sandwich Testing*. (4)

3 c) Study the *flowgraph* of a program on the next page. Create a *test table* with the minimum number of test cases for ensuring *branch coverage*. You can use the table tool in Wiseflow, to avoid using the camera.

What is the minimal number of test case sufficient for *full path coverage*? The answer is not obvious, so you need to write a motivation. (4)

If you decide to write on paper, take a photo of the table and add that as an appendix to the answer. Let the name of the appendix start with the characters "3c".



x, y, z are positive reals
:= means assignment
• is multiplication

Area 4: Planning and Processes

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

- A. A *project* can have several *goals*, but a *process* can only have one.
- B. A *toll-gate* must always be preceded by a *mile-stone*.
- C. It is fully possible to assign two or more people to a single activity in a *project plan*.
- D. A *project* occurs once, but a *process* can occur several times.

4 b) Describe one advantage and one disadvantage of having very short (1 week) *iterations* in *iterative development*. Also describe one advantage and one disadvantage for having very long *iterations* (12 weeks). (4)

4 c) Describe one similarity and one difference between SCRUM and Kanban. How is this difference motivated by the philosophies behind SCRUM and Kanban? (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. One of the most important parts of the CMMI process area Technical Solution (TS) is to develop and evaluate different approaches to the design of the product.
- B. One of the most important parts of the CMMI process area Requirements Management (REQM) is to elicit the true needs of all stakeholders.
- C. *Software review* is a technique that fits well in accomplishing the specific goals of the CMMI process area Verification (VER).
- D. You have reached a CMMI *maturity level* if you fulfil more than half of the *process areas* on that level.

5 b) Describe two different *metrics* or *measurements* that can be used to predict the *maintainability* of a software system. Don't forget to motivate why the *metric* or *measurement* can be an indicator. Remember the headlines: Description, how to obtain data, how to calculate the metric, and relevant quality factor. (4)

5 c) Describe the different phases of the PDCA cycle. You may use an example. PDCA = Plan Do Check Act. (4)

Part 2: Advanced

6. Originally *eXtreme Programming*, XP was composed of the following rules:



The Rules of Extreme Programming

Planning

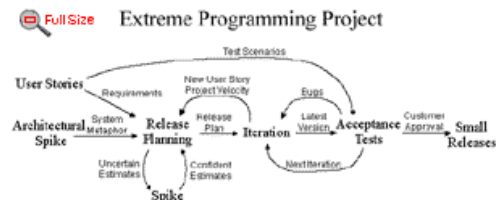
- User stories are written.
- Release planning creates the release schedule.
- Make frequent small releases.
- The project is divided into iterations.
- Iteration planning starts each iteration.

Managing

- Give the team a dedicated open work space.
- Set a sustainable pace.
- A stand up meeting starts each day.
- The Project Velocity is measured.
- Move people around.
- Fix XP when it breaks.

Designing

- Simplicity.
- Choose a system metaphor.
- Use CRC cards for design sessions.
- Create spike solutions to reduce risk.
- No functionality is added early.
- Refactor whenever and wherever possible.



Coding

- The customer is always available.
- Code must be written to agreed standards.
- Code the unit test first.
- All production code is pair programmed.
- Only one pair integrates code at a time.
- Integrate often.
- Set up a dedicated integration computer.
- Use collective ownership.

Testing

- All code must have unit tests.
- All code must pass all unit tests before it can be released.
- When a bug is found tests are created.
- Acceptance tests are run often and the score is published.

Suppose you are building a highly interactive software for the consumer market, for instance a mobile multiplayer game, in a team of eight skilled programmers. You are the team leader and have been asked to select five of the rules that you think would be the best for your project. State the five rules you select and for each rule, state:

- The name of the rule.
- A description of how it works.
- A description of expected benefits for your project.
- A short suggestion of how the effect of the rule can be evaluated.
- Potential drawbacks or risks of using the rule.

(20)

7. In several courses we use a system used to manage a gym, such as Campushallen, as a scenario for exploring different concepts in Software Engineering. The feature list is found in Appendix.

Now it is your task to make an overall *architecture* description of the system. You don't need to use UML; *box-and-line diagrams* will do. Hint: Don't be too detailed about the architecture, about 7 boxes will do. If you are not using UML, make sure to label all your boxes and arrows to clarify the relationship between the different elements.

Take a photo of the diagram and add that as an appendix to the answer. Let the name of the appendix start with the character "7".

In text, describe a quality factor that you think your architecture supports. Don't forget the motivation.

You are allowed to make assumptions beyond the feature list, as long as you write them down. (10)

8. From the gym system in Appendix, give examples of how the following *relationships* between *classes* in an *UML class diagram* can be used:

- a) Association
- b) Composition
- c) Generalization
- d) Realization
- e) Dependency

You don't need to draw a complete *UML class diagram*, just create examples of the *relationship* involving 2-3 *classes* each from a gym system.

Take a photo of the diagram and add that as an appendix to the answer. Let the name of the appendix start with the character "8". If you decide to use one diagram per page, name them 8a, 8b, etc.

You are allowed to make assumptions beyond the feature list, as long as you write them down. (10)

9. Use the gym system again, but this time you write five *functional requirements* of the system. For each requirement, describe how it can be tested. (10)

Appendix: Features of a gym system

- Membership management used by administrators to handle membership type, duration, different ways of payment, etc.
- Membership management used by personal trainers to set goals and follow up on progress. It also notifies if a member has been absent for more than two weeks.
- Member's app for viewing and updating personal data, including password. The app can be accessed on the web or installed on all major smart-phones and tablets. It's also possible to use an entrance card and log in at a computer in the reception.
- Communication module. Staff can send individual messages to members or broadcast to all.
- Chat function for members and staff on different topics.
- Automatic personal trainer. When a member using a smartphone app comes close to a work-out machine, it automatically displays preferences, e.g. height of seat, weight of resistance, number of repetitions, for that machine. After the exercise, the app asks you if you want to adjust anything for the next time. The result is stored with your progress data, which can be monitored by yourself and your personal trainer.
- Gold member's key. Gold members can open the front door and use the gym outside opening hours.
- Booking system for classes offered in the big hall.