

Written exam for Software Engineering Theory

Course codes TDDC88, TDDC93, 725G64

This is a special arrangement of an exam in Wiseflow. The intention is that all problems and solutions will be handled in Wiseflow. This manuscript is a back-up if there is a serious failure in Wiseflow and we need to continue with pen and paper.

Grading

The exam consists of two parts: Fundamental and Advanced.

The Fundamental part has problems worth 10 credits per area. Areas are Requirements, Design & Architecture, Testing & SCM, Planning & Processes, 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 comparable to 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 2020. 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.

Note that checking the wrong alternative in problems 1-5 a) gives minus credits. **This is a tougher requirement compared to exams given before the pandemic.**

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

Good Luck!

Kristian

Problems

Part 1: Fundamental

Area 1: Requirements

1 a) Which of the following statements are true? (Two statements are true. You get +1 for each correct answer chosen and -1 for each wrong answer chosen) (2)

- A. The requirement “When the user presses the OK-button, the system shall upload the file.” is *ambiguous* since it is not certain that a user will ever press the OK-button.
- B. The requirement “The system shall be able to upload the file within 60 seconds.” is a *non-functional requirement* since it specifies a quality factor.
- C. The requirement “When the user presses the Browse-button, the system shall display the file directory window.” is a *functional requirement*, since it specifies an input-output relationship.
- D. The requirement “Many users shall appreciate the commands possible to invoke.” is a *functional requirement*, even though it is not *testable*.

1 b) Scenario: You are developing an information system for an employment service. The system is web based and both prospective employees and employers can publish advertisements. Prospective employees can search and apply for jobs directly on the web and subscribe to newly entered advertisements matching their competence profiles. To use the service, the users need to register contact details to get a free account. The system administrators make sure that old advertisements and unserious advertisements are continually removed.

Task: Write at least two *use-cases* of the system with at least two different *actors* in total, e.g. one actor per use-case. (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: Design and Architecture

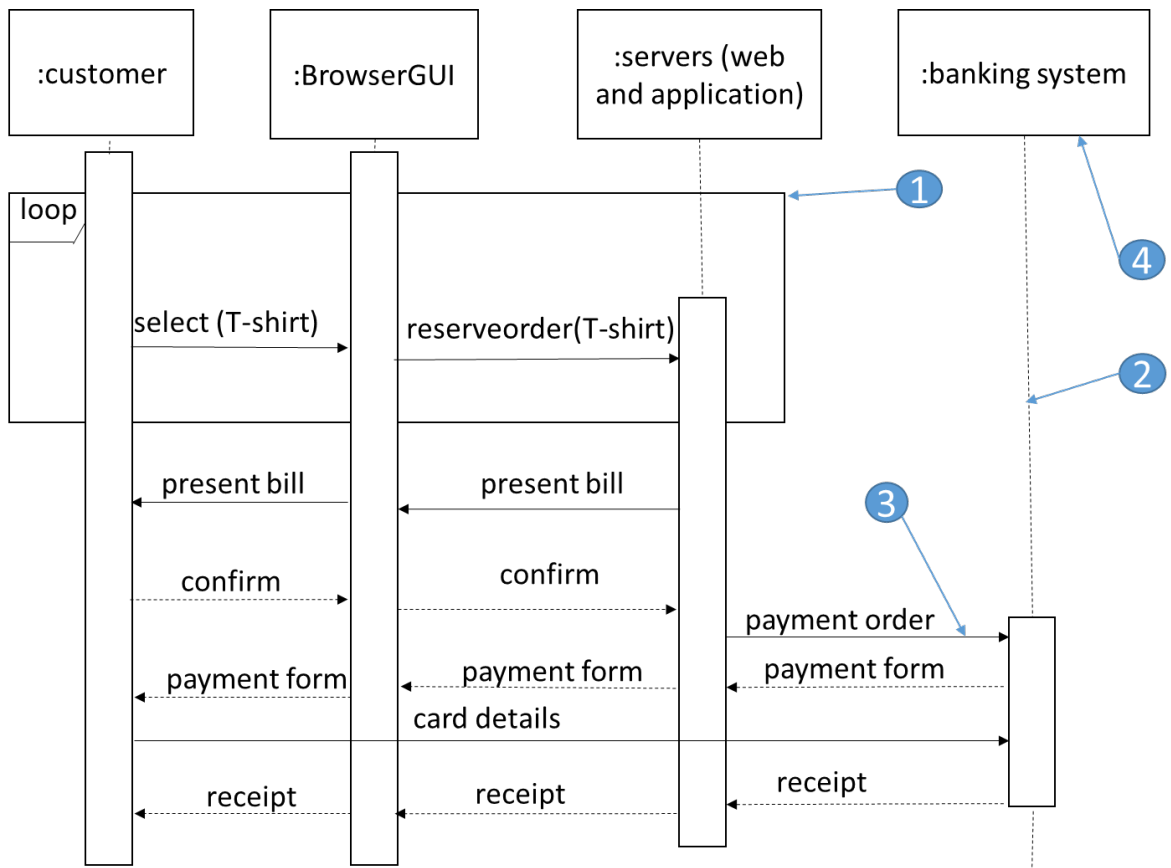
2 a) Which of the following statements are true? (Two statements are true. You get +1 for each correct answer chosen and -1 for each wrong answer chosen) (2)

- 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 thin client has comparably high demands on bandwidth in communication between client and server.

(2)

2 b) Suppose you have a *client-server architecture* and you get the feedback from your customers that the system is good, but that the *response time* is too long for everything that the system does. Suggest two different things you can do with the architecture to speed the system up. Don't forget the motivations
Hint: the problem does not specify which variant of client-server architecture you are using, so you have to make an assumption of which one. (4)

2 c) Modified in Wiseflow Describe the name of the model elements numbered in the *UML Sequence diagram* below. If you don't remember the names, describe the semantics of the elements (4)



Area 3: Testing and SCM

3 a) Which of the following statements are true? (Two statements are true. You get +1 for each correct answer chosen and -1 for each wrong answer chosen) (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.

3 b) Describe the concepts *regression testing*, *smoke testing*, *beta testing*, and *performance testing*. (4)

3 c) Describe the *feature-branch workflow* in Git which is an open-source distributed version control system. Mention one advantage and one disadvantage of this workflow. (4)

Area 4: Planning and Processes

4 a) Which of the following statements are true? (Two statements are true. You get +1 for each correct answer chosen and -1 for each wrong answer chosen) (2)

- A. In *Kanban* you must organize the work in *time-boxed sprints*.
- B. In *Kanban* you must have an upper limit of the number of items you can work with in parallel.
- C. In *SCRUM* you must measure the calendar time it takes from when you started working on an item, until you are finished.
- D. In *SCRUM* you must allow the *team* to select items for the *sprint* from the prioritized *product backlog*.

4 b) Describe two advantages and two potential problems of the (*classical*) *waterfall* life cycle model. (4)

4 c) Describe two *stakeholders* of the Lisam system which is used as course portal at LiU . Motivate why they are considered as *stakeholders* with 1-2 sentences. (4)

Area 5: Software Quality

5 a) Which of the following statements are true? (Two statements are true. You get +1 for each correct answer chosen and -1 for each wrong answer chosen) (2)

- A. The concept *non-personal software* derived from *total quality management* means that the software is embedded and not directly accessed by a human user.
- B. The principle *mutually beneficial supplier relationships* of *ISO 900-3* implies that you can take steps to help the supplier improve its processes.
- C. An organization at CMMI level 1 does not have a risk management process.
- D. An organization at CMMI level 3 typically has its standard processes that are adapted to the individual projects.

5 b) Describe 4 *roles* that participate in a *software inspection* process. For **each** role describe: i) What they do, and ii) What their responsibilities are. (4)

5 c) Scenario: Your newly started company is in the unusual situation that it develops many products from scratch. The customers are happy, but you have to spend lot of time to figure out viable solutions and processes, so you wish to be more efficient.

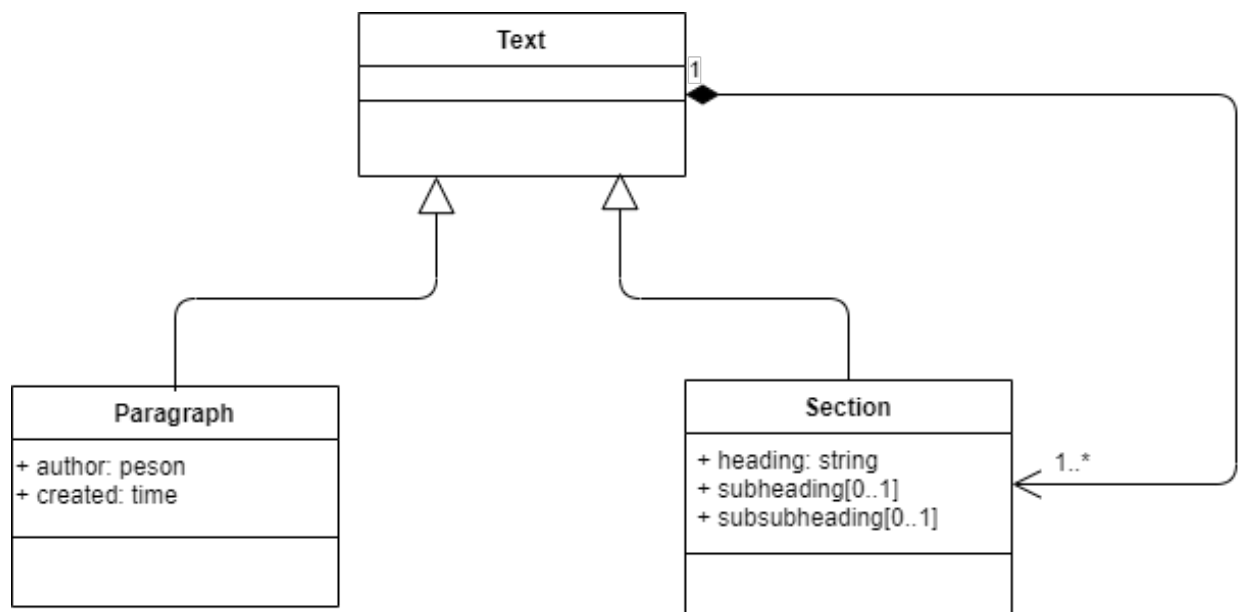
Task: Describe a *CMMI process area* that you think would help you to become more efficient. Summarize the process area's purpose and introductory notes in 5-6 sentences and motivate how this could help your company. (4)

Part 2: Advanced

6. *Scenario:* You are developing a text management system and want to analyse and visualize the following requirements:

1. A text can be a paragraph or a section.
2. A text knows its author and time of creation.
3. A section has a heading, and maybe a sub-heading.
4. A section contains at least one text.

Task: A junior designer showed you the *UML Class diagram* below. The diagram is far from correct. Write constructive feed-back to the junior designer with good explanations of how he/she should think in the future to avoid similar problems. (10)



7. *Scenario:* You are building a support system for a help desk that customers of broadband services can call if they have any problems. The personnel of the help desk have a one-week education and they use your system, which contains a decision tree and a searchable database of known cases. They can also share their screen with an expert that can be called during daytime. Administrative information about the user, program version, time, etc. are logged.

Task:

- a) Write a description of an activity you will use in the development project ensuring good system *usability*. Answer by describing: how the activity is performed, who are involved, what resources you need, when in the project the activity is performed. (8)
- b) Describe three different *metrics* that can be used to evaluate *usability*. For each metric, write: a description, how you obtain the data, how you calculate the metric, and how the metric can indicate good or bad *usability*. (12)

8. Identify four different *risks* of the development and/or operation of the helpdesk described in the scenario of problem 7. by selecting one risk per impact category: *insignificant, tolerable, serious, and catastrophic*. For each of the *risks*, write a short description of the risk and give an example of what you can do if it occurs. (10)

9. *Scenario*: Some of the course requirements for getting Master of Science Exam in Computer Science (*)

1. course requirements for a total of 120 ECTS credits of courses from the curriculum
2. passed the course TDDD89
3. courses on advancement level A (advanced) 90 ECTS credits including:
 - a. at least 30 ECTS credits courses from the major subject of Computer Science.
 - b. a 30 ECTS credits Master's Thesis in the major subject of Computer Science.

Suppose you have written a script that automatically reads applicable data from LADOK, validates the rules above, and either grant or deny the exam. In the case of denial, the number of the failing rule(s) is printed (e.g. 3a)

Task: Create a *test table* of at least five test-cases for *equivalence class testing* of your script given the rules above. You may assume that credits are non-negative integers. Add a column of applicable equivalence class(es) for each test case in the test table. (10)

(*) In real life, special curriculum related constraints might be added