

LTAT.05.003 – Software Engineering

Written Exam – 27 January 2020

Start: 14:15 – End: 16:45

Important Notes:

- You are only admitted to this exam, if you have at least 30 marks from the lab assignments.
- The exam is open-book and open-laptop. Web browsing is allowed, but you are neither allowed to use e-mail clients or Instant Messaging clients nor to share any information “live” with anybody inside or outside the exam room.
- This document (question sheet) contains 6 pages (including the cover page). Please check that you have received 6 pages.
- At the end of the exam you must submit both the question sheets and your answer sheets. To avoid that any of your solutions get lost, make sure to write your name (and student ID) on each sheet of paper that you submit.
- Write clearly. Answers that are illegible cannot be counted as correct answers. Only answers written in English will be marked.
- To answer Part 1, use the separately distributed answer sheet. Answers given on the question sheets will not be marked!
- To answer Parts 2 and 3, use the separately distributed blank paper. Answers given on the question sheets will not be marked! Also, please number the pages on your answer sheets.
- At the end of the exam you must return the problem sheet. If you take the question sheets with you (out of the exam room), this will be considered academic fraud (cheating) and treated accordingly.
- Total exam marks: 30 (equivalent to 30% of total course grade). You must get at least 10 marks in this exam to not fail the course.

=====

PART 1: Multiple-Choice Questionnaire (10 marks)

PART 2: Open Questions (10 marks)

PART 3: Constructive Tasks (10 marks)

Total: 30 marks (=100%)

PART 1: Multiple-Choice Questionnaire (10 marks)

Important: For Part 1, please check boxes on the separate answer sheet. Read carefully before you answer and observe instructions carefully!

The following questions (up to question Q-08) have exactly one correct answer, thus, you must check exactly one answer box on the separate answer sheet. If you think that more than one answer is correct, choose the one answer that seems to be most correct/suitable/relevant.

Question Q-09 at the end of Part 1 can but must not have exactly one correct answer. Please read related instructions carefully.

Q-01 (1 mark): Which of the following should apply to make the waterfall approach to software development a feasible option?

Answer choice:

- A: The business domain is well-known by the developers and not very dynamic
- B: There are few requirements and they are stable
- C: The product to be developed is not very innovative and the technologies used during development are well-established and not expected to change in the near future
- D: All of the above should apply

Q-02 (1 mark): What are User Story Points mainly used for?

Answer choice:

- A: To estimate the skills of developers
- B: To estimate project quality
- C: To improve the software development process
- D: None of the above are correct answers

Q-03 (1 mark): Which of the following is a performance requirement of a web-based application?

Answer choice:

- A: When the user clicks a "read me" link, the color of the link should change from pink to blue
- B: When the user clicks a "read me" link, the next page should be opened within 2 seconds
- C: When the user clicks a "read me" link, the 'mouse over' should show the target page title
- D: When the user clicks a "read me" link, the 'read me' page should load correctly

Q-04 (1 mark): A use case UC-Ex has an "extend" relationship to another use case UC-Base (i.e., the base use case). Which of the following statements is correct?

Answer choice:

- A: UC-Base is not complete without UC-Ex
- B: UC-Ex does not change the behavior of UC-Base
- C: The execution of UC-Ex is not conditional
- D: UC-Ex is optional

Q-05 (1 mark): Which of the following statements about code refactoring is correct?

Answer choice:

- A: Refactoring does always change the program design
- B: Refactoring is part of testing
- C: Before refactoring a component, a test suite for this component must be in place
- D: Code duplication is a refactoring

Q-06 (1 mark): Which of the following statements is **not** correct?

Answer choice:

- A: White-box testing techniques exploit knowledge about the code that is tested
- B: Regression testing cannot efficiently be done manually
- C: Equivalence class partitioning is a white-box testing technique
- D: Performance testing is usually done at systems level

Q-07 (1 mark): Which of the following statements applies best to the code smell “Speculative Generality“?

Answer choice:

- A: Redundancy in the naming of variables, methods and classes
- B: Developers (over-)generalize their code in an attempt to predict future needs
- C: A variable, parameter, method, code fragment, class, etc. is not used anywhere
- D: Classes with fields and getters and setters and nothing else

Q-08 (1 mark): Which of the following is **not** characteristic for Extreme Programming?

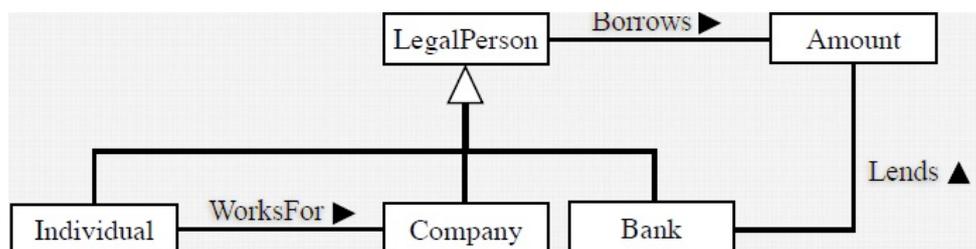
Answer choice:

- A: Task-boxing
- B: Time-boxing
- C: On-site customers
- D: Requirements that change over time

The following question (question Q-09) can have more (or less) than one correct answer. You must check all correct answer choices to get full marks. You get partial marks, if you check some of the correct answer choices. You will get a penalty, if you check an incorrect answer choice. You don't get a penalty, if you miss a correct answer choice. Overall, the lowest possible mark you can get is 0, i.e., even if everything you check is wrong, you won't get a negative mark.

Q-09 (2 marks): The figure below shows an initial (incomplete) sketch of a domain model for the following situation:

- An individual can work for one or several companies
- A car is owned by an individual, a bank, or a company (note: a bank is not a company)
- A bank gives an amount of money (a loan) for buying a car
- A loan can be secured against a car



The domain class ‘Car’ is missing in the model. If you add the missing class ‘Car’ to the model, with which other class(es) should it be associated?

Answer choice:

- A: Individual
- B: Bank
- C: Amount
- D: LegalPerson

PART 2: Open Questions (10 marks)

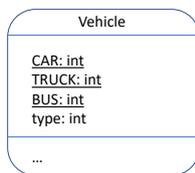
Note: Please give your answers on the provided blank answer sheet(s) and state clearly to which question number each answer refers. Answers that have no question number stated will not be marked. Don't forget to write your name on each separate answer sheet.

- Q-10** (3 marks): Assume that a list of requirements contains the following two user stories:
- (US1) As hotel IT administrator, I can indicate folders not to backup so that my backup drive isn't filled up with data I don't need to be saved
- (US2) As hotel receptionist, I want to see the list of rooms not yet booked so that I can deduce the number of booked rooms

Answer the following questions a) to d):

- Which of the two user stories is of better quality? Justify your answer.
- What are the three elements of a complete user story? Explain the purpose of each part briefly.
- Explain how the 'why'-part of a user story is helpful to the Product Owner in a SCRUM project.
- Is the 'why'-part also important for the SCRUM team? Justify your answer.

Q-11 (4 marks): Type code occurs when, instead of a separate data type, you have a set of numbers or strings that form a list of allowable values for some entity. Often these specific numbers and strings are given understandable names via constants, which is the reason for why such type code is encountered so much. Typical examples of type code are shown in the example class:

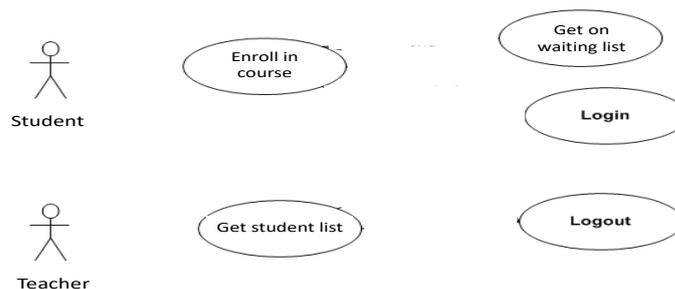


Do the following:

- In object-oriented programming, type code is considered a code smell. Using the Vehicle class shown above, explain the meanings of two possible refactoring options called **'Replace Type Code with Class'** and **'Replace Type Code with Subclasses'**
- Assume that in a program having the Vehicle class shown above, the coded type is used in a switch statement and directly affects the program's behavior. Answer the following: b1) Which refactoring would you choose? Justify your answer. b2) How does the refactored UML class diagram of Vehicle look like?
- Assume you consider applying refactoring 'Replace Type Code with Subclasses' but then you notice that your program allows a Truck object to be converted into a Bus object. Can you still apply the intended refactoring? (justify your answer)

Q-12 (3 marks): Do the following:

- A use case diagram contains **actors** and **use cases**. Give a one-sentence definition of each.
- Explain the difference between primary and secondary actors and give an example of each (Hint: you must give the actor examples in the context of an example use case).
- In a use case diagram use cases can be included in other use cases (= base use cases). Give two conditions that must be fulfilled for a use case to be included in a base use case.
- Explain what an 'alternate scenario' in a use case description is and give an example.
- Copy the figure shown below and connect the actors and use cases properly. If labels <<include>> or <<extend>> make sense, add them on the connecting lines.



PART 3: Constructive Tasks (10 marks)

Note: Please provide solution on the provided blank answer sheet(s) and state clearly to which task number each solution relates. Solutions that have no task number stated will not be marked. Don't forget to write your name on each separate answer sheet.

T-01 (3 marks): Assume, you have 20 user stories (US1-US20) in your project backlog. You have estimated the user stories to have a difficulty/complexity expressed in story points as follows:

- Each of US1 to US5 equals 2 story points
- Each of US6 to US10 equals 3 story points
- Each of US11 to US15 equals 6 story points
- Each of US16 to US20 equals 14 story points

To do:

Q1: If you have a team of 4 developers and weekly sprints (1 week = 5 days = 40 hours), which user stories would you be able to implement in the next sprint and achieve the highest possible value without violating your capacity (effort) constraint?

Q2: How would your result change if the following set of dependencies among user stories applies?

- US6 must be implemented together with US11
- US7 must be implemented together with US16
- US8 must be implemented together with US17

How much is the overall value affected in the solution of Q2?

For your calculations, assume the following:

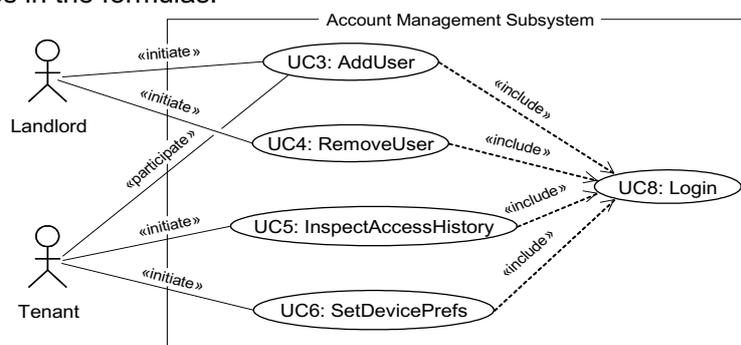
- All user stories have equal value for the end user.
- In past projects, on average, one developer could implement 1% of the sum of user story points of your current project in one day. In the current project, developers have the same productivity as in the past.
- We don't allow overtime.

Show all your calculations and explain all your choices. (If you just present the lists of User Stories that can be accommodated in Q1 and Q2, you will get 0 marks, even if the lists are correct.)

T-02 (4 marks): Make the following assumptions for the elements in the use case diagram shown in the figure below:

- All actors interact with the system via GUIs
- UC3 and UC5 have 5 transactions each
- UC4 has 4 transactions
- UC6 and UC8 have 2 transactions each
- The 'Usability' and 'Reusability' of the system extremely important. Special User Training Facilities are not needed. Each of the other Technical Factors equals '2'
- Each of the eight Environmental Factors (EF(1) ... EF(8)) equals '3'

Calculate the Unadjusted Use Case Points (UUCP), use Case Points (UCP) and Project Effort for the use case diagram in the figure below. Show all your calculations. If you only write down end results without showing the formulas used and calculations made, you won't get marks. Also, you must justify all values for the variables in the formulas.



T-03 (3 marks): The code snippet below shows a function that calculates fines for speeding when driving a car. Fines are calculated based on the age of the driver (age in years), the excess speed (overspeed in km/h), and the speed limit (speedlimit in km/h).

```

0 public static int speedingfine (int age, int overspeed, int speedlimit) {
1     int fine = 0;
2     if ((age >= 25) && (overspeed < 20) && (speedlimit >= 50)) {
3         fine = fine + 10 * overspeed; }
4     if ((age < 25) || (speedlimit < 50)) {
5         fine = fine + (30 * overspeed); }
6     if (overspeed >= 20) {
7         fine = fine + 500; }
8     return fine; }

```

To do:

Provide one single minimum set of test cases that achieves both (a) 100% statement and (b) 100% decision coverage. Remember that complete test cases include both input values and expected output values. You must also say for each test case which lines it covers, which decisions it covers, and in which direction it covers each covered decision (i.e., 'true' or 'false').

Note: You will get a penalty, if the number of test cases is not the minimum. You can assume that all inputs have been checked for syntactic/semantic correctness, i.e., all parameter values are of the correct type and in a reasonable range (e.g., age is in [18, 99]).

Hint: Use the following table format to present your test cases of part a) and part b):

Test case number	Age (in)	Overspeed (in)	Speedlimit (in)	Fine (out)	a) Lines covered b) Decision(s) covered (T/F)
1					
...					
...					