

## LTAT.05.003 – Software Engineering

Written Exam – 09 January 2018

Start: 14:15 – End: 16:45

### Important Notes:

- 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. There are 2 bonus marks available.

=====  
**PART 1: Multiple-Choice Questionnaire (10 marks + 2 bonus marks)**

**PART 2: Open Questions (10 marks)**

**PART 3: Constructive Tasks (10 marks)**

---

**Total: 30 marks (=100%) plus 2 bonus marks**

## **PART 1: Multiple-Choice Questionnaire (10 marks + 2 bonus 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-10) 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-11 at the end of Part 1 is a bonus questions. Please read related instructions carefully.

**Q-01** (1 mark): Which of the following indicate that a waterfall approach to software development is not appropriate?

Answer choice:

- A: Business is volatile
- B: There are few requirements
- C: Business requirements are stable
- D: Legal regulations require completing requirements analysis before starting software design

**Q-02** (1 mark): Which of the following is not correctly characterizing an <<include>> use case?

Answer choice:

- A: This use case is optional
- B: The execution of this use case is not conditional
- C: The base use case is not complete without this use case
- D: This use case does not change the behavior of the base use case

**Q-03** (1 mark): What are User Story Points typically used for?

Answer choice:

- A: To estimate project effort
- B: To estimate project quality
- C: To estimate project duration
- D: both A and C are correct answers

**Q-04** (1 mark): Which of the following is a non-functional 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 blue to pink
- B: When the user clicks a "read me" link, the next page should be opened within 5 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-05** (1 mark): Which of the following are the key elements of use case diagrams?

Answer choice:

- A: People, computer
- B: Actors, use cases
- C: People, classes, objects
- D: Uses, use cases

**Q-06** (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 a test activity
- C: Before refactoring a component, a test suite for this component must be in place
- B: Refactoring is done in order to add new functionality

**Q-07** (1 mark): Which of the following statements is correct?

Answer choice:

- A: Regression testing can efficiently be done manually
- B: Equivalence class partitioning is a white-box testing technique
- C: Black-box testing techniques exploit knowledge about the code that is tested
- D: White-box testing techniques exploit knowledge about the code that is tested

**Q-08** (1 mark): You execute a calculation program and see the following message on the screen: 'Program aborted due to division by 0'. Which of the following describes your experience best?

Answer choice:

- A: I made an error (while using the program)
- B: I debugged an error
- C: I triggered a failure
- D: I localized a fault

**Q-09** (1 mark): Which of the following statements applies best to the code smell 'Speculative Generality'?

Answer choice:

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

**Q-10** (1 mark): If three developers work three days each, how much effort do they spend?

Answer choice:

- A: 9 days
- B: 9 person-days
- C: 3 person-days
- D: 3 days

The following question (question Q-11) is a bonus question and can have more 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-11** (2 marks - bonus): Which of the following is a characteristic of SCRUM?

Answer choice:

- A: Task-boxing
- B: Agile nature
- C: Stable requirements
- D: Time-boxing

## PART 2: Open Questions (10 marks)

Note: Please give your answers on separate 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.

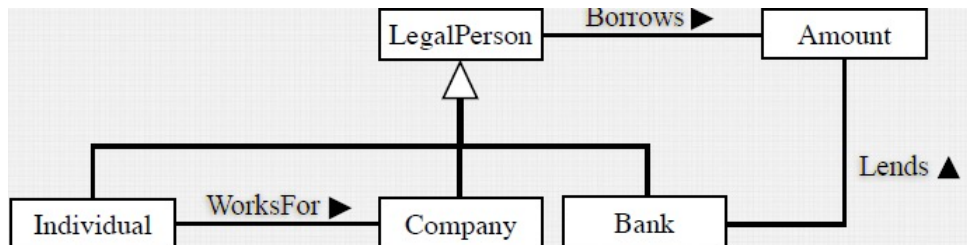
**Q-12** (4 marks): Assume that a list of requirements contains the following two user stories:  
(US1) As a hotel receptionist, I want to see the list of rooms not yet booked.  
(US2) As a user, I can indicate folders not to backup so that my backup drive isn't filled up with things I don't need saved.

Answer the following questions Q1 to Q3:

- Q1: Which of the two user stories is of better quality? Justify your answer.
- Q2: What are the three elements of a complete user story? Explain each part briefly
- Q3: What is the purpose of the 'why'-part in a user story: For whom and how is it useful?

**Q-13** (3 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
- Banks give loans (loan = amount of money) for buying cars
- A loan can be secured against a car



To Do:

- Say which domain class is missing (only using the information given).
- Place the missing domain class in the domain model, i.e., show the association(s) required to connect the missing domain class properly with the existing domain class(es) in the model. Provide name and direction of the added association(s).

**Q-14** (3 marks): Lean software development recognizes seven types of waste. A particularly bad waste is “partially done work”. Answer the following questions related to “partially done work”:

(a) For each of the following items say whether it is “partially done work”. If you consider an item to be “partially done work” explain why this item constitutes waste. [2 marks]:

1. Commented out code
2. Undocumented code
3. Completed, tested, and deployed code

(b) Give an additional example of “partially done work” (briefly explain why it is waste – without explanation you won't get a mark for the given example, even if it is waste of the type “partially done work”)

### PART 3: Constructive Tasks (10 marks)

Note: Please provide solution on separate 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 dependencies between user stories apply?

- 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** (3 marks): Point out the code smell in the code shown below. Then refactor the code. Show the refactored code and briefly explain what you did (i.e., explain how the refactored code is better than the original code).

```
double getPayAmount() {
    double result;
    if (isDead) result = deadAmount();
    else {
        if (isSeparated) result = separatedAmount();
        else {
            if (isRetired) result = retiredAmount();
            else result = normalPayAmount();
        }
    }
    return result;
}
```

**T-03 (4 marks):** Make the following assumptions for the elements in the use case diagram shown in the figure below:

- Actors are of type 'complex'
- UC3 and UC5 have 5 transactions each
- UC4 has 4 transactions
- UC6 and UC8 have 2 transactions each
- The Technical Complexity Factor equals '1'
- All eight Environmental Factors (EF(1) ... EF(8)) equal '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.

