MTAT.03.094 – Software Engineering

Written Exam – 08 January 2016

Start: 14:15 – End: 16:15

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.
- 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. Also, please number the pages on your answer sheets.
- Write clearly. Answers that are illegible cannot be counted as correct answers. Only answers written in English will be marked.
- 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 5 bonus marks available.

==========

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

PART 2: Open Questions (10 marks + 3 bonus marks)

PART 3: Constructive Tasks (10 marks)

Total: 30 marks (=100%) plus 5 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 statements does not describe a good work practice of a software engineer?
Answer choice:
A: To adhere to the existing body of knowledge of software development techniques/methods
B: To observe constraints, e.g., time and budget constraints, and standards
C: To skip a quality assurance activity (e.g., system testing), because a deadline is approaching
D: To develop and use design and quality models

Q-02 (1 mark): Which of the following statements does not explain why requirements engineering is difficult?
Answer choice:
A: Because no UML notation exists for describing requirements
B: Because requirements can change over time
C: Because it is difficult to identify the relevant stakeholders, and, once identified, the stakeholders have difficulties describing what they want/need
D: Because stakeholders have conflicting requirements

Q-03 (1 mark): What are User Story Points not used for?
Answer choice:
A: To estimate project effort
B: To estimate project quality
C: To estimate project duration
D: To estimate team velocity

Q-04 (1 mark): Which of the following statements about use case descriptions is not correct?
Answer choice:
A: A participating actor is an actor who helps achieve the goals of the initiating actor
B: An alternate flow describes exeptions from or extensions to the normal interaction scenario
C: Preconditions describe the state of the system before the start of the interaction scenario
D: A use case diagram is a graphical representation of a use case description

Q-05 (1 mark): Which of the following statements about UML class diagrams is not correct?
Answer choice:
A: Inheritance is a specific type of composition relationship
B: A class has the following elements: name, list of attributes, list of operations
C: Composition describes a whole/part relationship
D: A superclass and its subclasses describe a generalization relationship

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 done in order to add new functionality
C: Refactoring is a test activity
D: Before refactoring a component, a test suite for this component must be in place

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 types of meetings is not defined by SCRUM?
Answer choice:
A: SCRUM Master Meeting
B: Sprint Retrospective
C: Daily Scrum (or 'Stand-Up Meeting')
D: Sprint Planning Meeting

Q-10 (1 mark): If three developers work 2 days each, how much effort have they spent?
Answer choice:
A: 6 person-days
B: 6 days
C: 2 person-days
D: None of the above

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): You plan to test the correct functioning of a text editor's 5 font effects: bold, italics, underline, shadow, and strikethrough. Each effect can be 'on' or 'off'. What is correct below?
Answer choice:
A: Complete testing of all possible combinations of font effects requires 32 test cases
B: The total number of pairwise interactions among font effects is 40
C: Complete testing of pairwise interactions can be done with only 4 test cases
D: Complete testing of all 5-way interactions of font effects requires less than 32 test cases

PART 2: Open Questions (10 marks + 3 bonus 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 (2 marks): 'Software Engineering' (SE) is an engineering approach to industrial software development. As in other engineering disciplines, modeling and standardization play an important role in SE. Do the following tasks T1 to T3:

- T1: Give a concrete example of a modeling language used in SE
- T2: Give a concrete example of an international standard used in SE
- T3: Explain why every good software engineer must also be a good software craftsman but not every good software craftsman needs to be a good software engineer

Q-13 (3 marks): Assume that a list of requirements contains the following two user stories:

(US1) As a user, I want to narrow down city search results by country, so I can find the right city more quickly.

(US2) As a hotel receptionist, I want to see the list of rooms not yet booked

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 why is it useful?

Q-14 (2 marks): Assume you use an open source spreadsheet program to do simple arithmetic. You only use cells in row 1 of the worksheet. You want to multiply two numbers which you enter in the cells of columns 1 and 2, and you want the result displayed in the cell of column 3. In addition, you want the result shown in column 3 multiplied by 2 and displayed in the cell of column 4. You define the required functions for cells 3 and 4. Then you test the functions implemented in the spreadsheet with 5 tests, i.e., you enter two numbers in columns 1 and 2 and get the following results in cells 3 and 4:

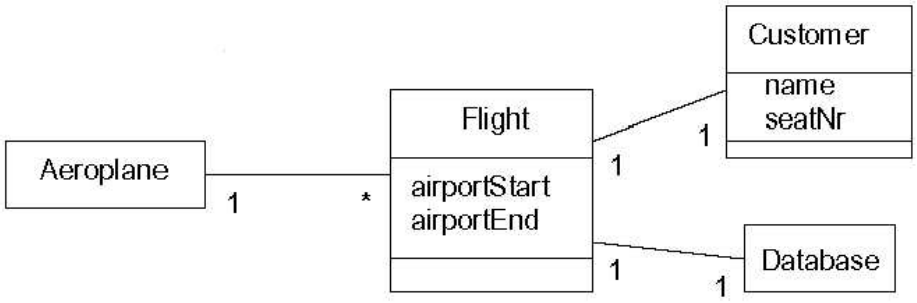
- 1st test: 3 11 → 33 66
- 2nd test: 3 12 → ## 72
- 3rd test: 12 7 → 84 168
- 4th test: 10 9 → 90 180
- 5th test: 10 4 → ## 80

Obviously, the spreadsheet program doesn't work correctly, because the cells showing '##' should actually display the values '36' and '40'. Answer the following questions:

1. Did you observe one or two failures? If you observed two failures, were they of the same type? Justify/explain your answers
2. What could possibly be the cause of the observed failure(s)? Try to make your speculation about the possible fault(s) plausible, i.e., use only the information given in the problem statement to justify your speculation.

Q-15 (3 marks): The domain model shown below (next page) describes an incomplete – and incorrect – sketch of a booking system of an aviation company. Pick 3 of the 4 elements listed below and explain what is wrong with each of them (hint: do not add information – look for faults in the given information!):

1. Association between domain classes 'Flight' and 'Customer'
2. Domain class 'Database'
3. Attributes in domain class 'Customer'
4. Attributes in domain class 'Flight'



Q-16 (3 marks – bonus): Point out the code smells in the code shown below. Then briefly describe what refactoring steps you would take to refactor the code. Note: You don't need to write down the refactored code, it's sufficient, if you say how you refactor the code to remove the code smells.

```
class BookRental {
   String bookTitle;
   String author;
   Date rentDate;
   Date dueDate;
   double rentalFee;
   boolean isOverdue() {
      Date now=new Date();
      return dueDate.before(now);
   }
   double getTotalFee() {
      return isOverdue() ? 1.2*rentalFee : rentalFee;
   }
}
class MovieRental {
   String movieTitle;
   int classification;
   Date rentDate;
   Date dueDate;
   double rentalFee;
   boolean isOverdue() {
      Date now=new Date();
      return dueDate.before(now);
   }
   double getTotalFee() {
      return isOverdue() ? Math.max(1.3*rentalFee, rentalFee+20) : rentalFee;
   }
}
```

PART 3: Constructive Tasks (10 marks)

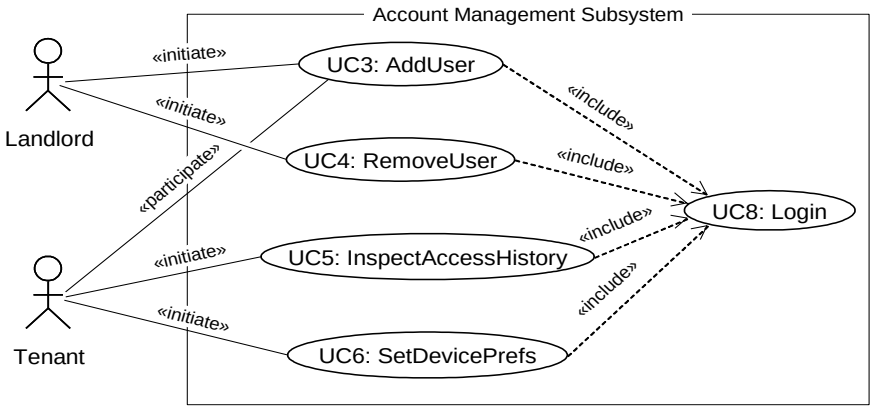
Note: Please provide solution on separate answer sheet(s) and state clearly to which task number each solution relates. Solutions that have no task number stated will not be marked. Don't forget to write your name on each separate answer sheet.

T-01 (3 marks): Three persons A, B and C independently review the same design specification. Person A detects 8 faults. Person B detects 10 faults. 4 of the faults found by B are also in the set of the 8 faults found by A. Person C detects 10 faults. 7 of the faults found by C have already been detected by either A or B. Estimate how many faults are still in the design specification after all faults found by A, B and C have been corrected. (Assume that no new faults are introduced when corrections are made.)
To get full marks you must show all your calculations. If you only write down a number for the estimated faults, you will get 0 marks (even if the number you come up with is plausible).

T-02 (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 UC4 have 5 transactions each
- UC5 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.



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 age of the driver (age), the excess speed (overspeed), and the number of penalty points (penaltypts) already accumulated in the driver's police record.

```
1  public static int speedingfine (int age, int overspeed; int penaltypts) {
2    int fine = 0;
3    if ((age >= 25) && (overspeed < 30) && (penaltypts < 3))
4       fine = fine + 100 * overspeed;
5    if ((age < 25) || (penaltypts >= 3))
6       fine = fine + (200 * overspeed);
7    if (overspeed >= 30)
8       fine = fine + 5000;
9    return fine; }
```

To do:

Provide a minimum set of test cases that achieves 100% statement coverage. Remember that complete test cases include both input values and expected output values. (Note: You will get a penalty, if the number of test cases is not the minimum)

Hint: Use the following table format to present your test cases:

Test case number	Age (in)	Overspeed (in)	Penaltypts (in)	Fine (out)
1				
...				
...				