MTAT.03.094 – Software Engineering

Written Exam – 10 January 2014

Start: 9:15 – End: 11:45

Important Notes:

- The exam is open-book and open-laptop. Web browsing is allowed, but you are not allowed to use e-mail clients nor 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 marks: 30 (equivalent to 30% total grade). You must get at least 10 marks in this exam to not fail the course.

==

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)

Note: For Part 1, please check boxes on the separate questionnaire 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.

Q-01 (1 mark): Which of the following statements describes work that is not part of the (core) work of a software engineer?
Answer choice:
A: To understand the problem domain of a software system to-be-developed
B: To define/evolve the architecture of a software system
C: To design and develop program code
D: To invent a new programming language

Q-02 (1 mark): For which of the following requirements engineering activities is the UML notation the least suitable/useful?
Answer choice:
A: Requirements gathering/elicitation
B: Requirements analysis
C: Requirements specification
D: Requirements validation (e.g., consistency checking)

Q-03 (1 mark): Which of the following statements about UML class diagrams is not correct.
Answer choice:
A: A class has the following elements: name, list of attributes, list of operations
B: Aggregation is a specific type of inheritance relationship
C: Composition describes a whole/part relationship
D: A superclass and its subclasses describe a generalization relationship

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: A use case diagram is a graphical representation of a use case description
D: Preconditions describe the state of the system before the start of the interaction scenario

Q-05 (1 mark): Why is requirements elicitation difficult?
Answer choice:
A: Because there doesn’t exist a suitable UML notation for requirements elicitation
B: Because it is difficult to identify the relevant stakeholders, and, once identified, the stakeholders have difficulties describing what they want/need
C: Because requirements can change over time
D: Because stakeholders don’t understand use cases

Q-06 (1 mark): Which of the following statements about code refactoring is correct?
Answer choice:
A: Before refactoring a code module, a test suite for this code module must be in place
B: Refactoring is done in order to add new functionality
C: Refactoring is a test activity
D: Refactoring does not influence (or change) the program design

Q-07 (1 mark): Which of the following statements about test techniques is correct?
Answer choice:
A: Mutation testing does require a test oracle derived from the program specification
B: Equivalence class partitioning is a white-box testing technique
C: White-box testing techniques exploit knowledge about the code that is tested
D: Black-box testing techniques exploit knowledge about the code that is tested

Q-08 (1 mark): You execute a program and you see the following message on the screen: 'Program aborted due to division by 0'. Which of the following terms describes your experience best?
Answer choice:
A: I made an error
B: I detected an error
C: I detected a fault
D: I triggered a failure

Q-09 (1 mark): Which of the following types of meetings <u>is not</u> defined by SCRUM?
Answer choice:
A: Product Owner Meeting
B: Sprint Retrospective
C: Daily Scrum
D: Sprint Review

Q-10 (1 mark): Which of the following is <u>not</u> a suitable unit for effort measurement?
Answer choice:
A: Person-Day
B: Week
C: Man-Month
D: Developer-Hour

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 essential for XP (Extreme Programming) but <u>not</u> for SCRUM?
Answer choice:
A: A role model
B: Pair-Programming
C: Test-Driven Development
D: Sprint Burn-down Chart

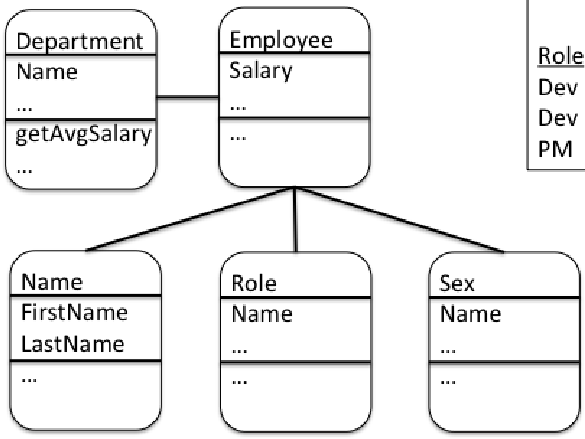
PART 2: Open Questions (10 marks + 3 bonus marks)

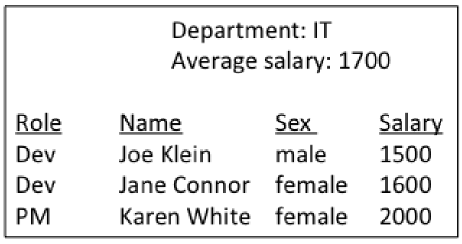
Note: Please give your answers on separate answer sheet(s) and state clearly to which question number each answer refers. Answers that have no question number stated will not be marked. Don't forget to write your name on each separate answer sheet.

Q-12 (3 marks): Briefly explain (point out) the most important differences between the following:
(1) Software Engineering -- Programming
(2) Software Engineering -- Software Development in a Craftsmanship-style
(3) Software Engineering -- Systems Engineering

Q-13 (3 marks): First explain why it is important to distinguish between 'error', 'fault' and 'failure'. Then give an example of each of the three concepts (i.e., three examples in total). In you examples, make it clear how the 'error', 'fault', and 'failure' you are describing either is introduced/committed, or detected/encountered.

Q-14 (2 marks): In the following excerpt of a domain model, there is (at least) one obvious mistake. Spot the mistake and briefly explain why it is a mistake. To get two marks, it is sufficient that you point out one mistake and explain it – even if you think that there is more than one mistake in the domain model.
Note: The excerpt of the domain model below on the left is intended to model (a part of) a payroll system that – among other things – will produce reports similar to the one shown below on the right.




Department: IT
Average salary: 1700

Role	Name	Sex	Salary
Dev	Joe Klein	male	1500
Dev	Jane Connor	female	1600
PM	Karen White	female	2000

Q-15 (2 marks): State one type of waste that is reduced by implementing each recommendation listed below (Note: the recommendations are made for agile projects using SCRUM):

- Recommendation 1: One person always works on one task at a time.
- Recommendation 2: The items in the product backlog are prioritized (and re-prioritized if needed) and only refined as they move up the priority scale.

Q-16 (3 marks – bonus): Assume, you hear the following sentence: „We decided to measure code size by measuring the length of Java classes as Lines of Code (LOC)." What is the correct mapping of the four underlined terms to the following elements of the described measure: Name (1), Entity (2), Attribute (3), and Unit (4)?

Note: You will get a penalty, if you incorrectly assign an underlined term to any of (1) to (4). For not assigning a term to any of (1) to (4), you won't get a penalty. Also, in case you give more wrong than correct answers, the worst thing that can happen is that you get 0 marks.

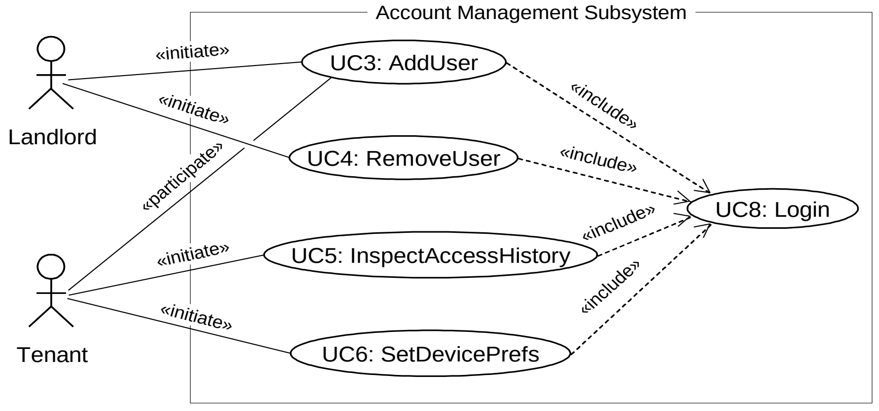
PART 3: Constructive Tasks (10 marks)

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): 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

Calculate the Unadjusted Use Case Points (UUCP) for the use case diagram in the figure below. Show all your calculations and briefly explain each of your steps.

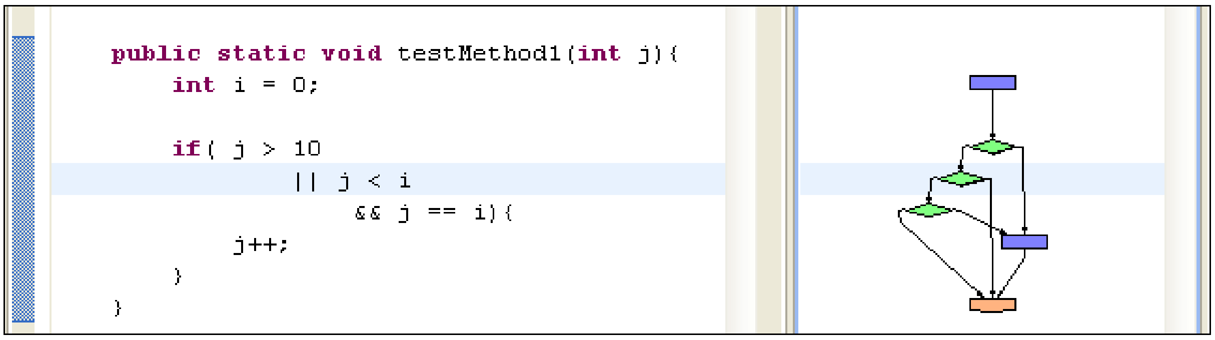


T-02 (5 marks): The figure below shows a little java code snippet on the left with a corresponding automatically generated control flow graph (CFG) on the right. Note that the if-statement is decomposed into its elementary conditions, i.e., 'j > 10', 'j < i', and 'j == i'. Do the following:

(1) Map the two value assignments ('i = 0' and 'j++') as well as the three elementary conditions ('j > 10', 'j < i', 'j == i') to the elements in the CFG. Note that there is exactly one correct mapping. (2 of 5 marks)

(2) Calculate the Cyclomatic Complexity (McCabe). Show your calculation. (1 of 5 marks)

(3) Design a set of test cases that will cover all linearly independent paths in the CFG. Is this possible? If not, try to cover as many paths as possible and say which path(s) cannot be covered (providing a reason why you think it cannot be covered). Note that each test case must specify the value of j when passed to testMethod1 and the value of j at the end of testMethod1. (2 of 5 marks)



```
public static void testMethod1(int j){
    int i = 0;

    if( j > 10
            || j < i
                && j == i){
        j++;
    }
}
```

T-03 (2 marks): Two persons independently review the same requirements specification. Person A detects 6 faults. Person B detects 8 faults. 4 of the faults found by B are also in the set of the 6 faults found by A. Estimate how many faults are still in the document after all faults found by A and B have been corrected. (Assume that no new faults are introduced when corrections are made.)
To get full marks you must show all your calculations and explain what estimation (or prediction) model you are using and what assumptions you are making. 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).