

## MTAT.03.159 – Software Testing

Written Exam – 30 May 2013

### Important Notes:

- The exam is open-book and open-laptop. Web browsing is allowed, but you are not allowed to use e-mail clients or Instant Messaging clients, or 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.
- Please state clearly at the beginning of your answer first sheet whether you plan to graduate in June. In that case your exam will definitely be graded before the end of June 5.
- Write clearly. Answers that are illegible cannot be counted as correct answers. Only answers written in English will be marked.
- Total marks: 40 (equivalent to 40% of total grade) plus 3 bonus marks.

=====

### Task 1 (8 marks):

a) Define the terms *error*, *failure* and *fault*. Give an example of each.

(3 marks)

b) Put the three terms in right place in the following sentence.

*A programmer made a(n) \_\_\_\_\_ which resulted in a(n) \_\_\_\_\_ in the program code, which when executed triggered a(n) \_\_\_\_\_.*

(1 mark)

c) The table below shows three terms in column 1 and three phrases in column 2. Each phrase in column 2 characterises exactly one term in column 1. Identify the correctly matching (term, phrase)-pairings.

| Term (Column 1)  | Phrase (Column 2)  |
|------------------|--|
| 1 - Verification | A - Fault localisation and repair  |
| 2 - Validation   | B - Check whether the software has been done right (i.e., software is working as specified)                        |
| 3 - Debugging    | C - Check whether the right software has been developed (i.e., software is working as needed/expected by customer) |

(2 marks)

d) Give two examples of activities that are primarily used for verification and two examples of activities that are primarily used for validation. An activity can be, for example, unit test, integration test, system test, regression test, stress test, load test, acceptance test, usability

test, certification test, requirements inspection, design inspection, code inspection, and so on. Briefly explain how each of the activities you picked support verification or validation.

(2 marks)

**Task 2 (4 marks):**

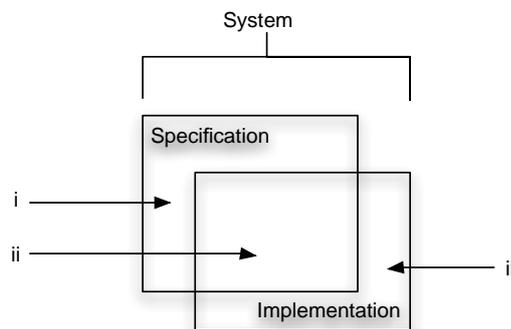
Briefly describe the essentials of *white-box* and *black-box testing strategies*. In addition, give two examples for each testing strategy (i.e., four examples of test strategies in total).

(4 marks)

**Task 3 (3 marks):**

The figure below shows an example of a system with a mismatch between its specified and implemented functionality. Which sets of functionality, (i), (ii) and (iii), may be tested by *white-box* and *black-box testing strategies*, respectively. Motivate/justify your answer.

(3 marks)



**Task 4 (14 marks):**

For the method `speedingfine` below, perform tasks a), b) and c):

```

1 public static int speedingfine (int age, int overspeed; int
licencemark) {
2     int fine = 0;
3     if ((age >= 25) && (overspeed < 30) && (licencemark < 3)) {
4         fine = fine + 100 * overspeed;
5     }
6     else {
7         if ((age < 25) || (licencemark >= 3)) {
8             fine = fine + (200 * overspeed);
9         }
10        if (overspeed >= 30) {
11            fine = fine + 5000;
12        }
13    }
14    return fine;
15 }

```

a) Draw the control flow graph and calculate the McCabe Cyclomatic number (i.e., the number of linearly independent paths) (2 marks)

b) Define test cases needed to achieve 100% coverage of linearly independent paths. Say explicitly which path is covered by each test case. (6 marks)

c) Set up the def-use table for all four variables. To make the task easier, the following table shows three example rows of the def-use table you are supposed to set up. Once you have completed the def-use table, define test cases that achieve 100% def-use coverage. Say explicitly which def-use paths are covered by each test case.

| Variable | Def | Use      | Du-path                       |
|----------|-----|----------|-------------------------------|
| Age      | 1   | 3, 7     | 1-3 (A1), 1-3-7 (A2)          |
| ...      |     |          |                               |
| ...      |     |          |                               |
| Fine     | 2   | 4, 8, 11 | 2-4 (D1), 2-8 (D2), 2-11 (D3) |
| ...      |     |          |                               |
| ...      |     |          |                               |
| Fine     | 11  | 14       | 11-14 (D7)                    |

(6 marks)

Remember that complete test cases include both input values and expected output values.

Task 5 (7 marks):

An industrial controller for process automation may be configured in a number of ways, changing the values of the four parameters shown below. Note that one parameter has 3 values while the other three parameters have only 2 values.

| Parameter         | Value 1                | Value 2                   | Value 3 |
|-------------------|------------------------|---------------------------|---------|
| Safety level      | Non-SIL                | SIL1                      | SIL2    |
| CPU               | 200 Mhz                | 400 Mhz                   |         |
| Redundant CPUs    | Single node controller | Two redundant controllers |         |
| Communication bus | Module bus             | Optic bus                 |         |

a) If you wanted to test all possible combinations of parameter values (i.e., the number of all 4-way interactions), how many test cases would you need? Show your calculation.

(1 mark)

b) How many different parameter pairs exist? Show your calculation.

(1 mark)

c) How many pairwise interactions (i.e., 2-way interactions) between parameter values exist? Show your calculation.

(1 mark)

d) How many 3-way interactions between parameter values exist? Show your calculation.

(1 mark)

e) Show that you can cover all 2-way interactions between parameter values with only 6 test cases. Present the set of test cases (per test case you need to specify only the values of the configuration parameters)

(3 marks)

Task 6 (1 mark):

Two reviewers have inspected a document and found the defects shown in Table 1 below. The Lincoln-Peterson model to estimate the total number of defects uses the following formula:

$N = n1 * n2 / n$       where  
N = estimated total number of defects  
n1 = number of defects found by reviewer 1  
n2 = number of defects found by reviewer 2  
n = number of defects found by both reviewers

| Defect | Reviewer 1 | Reviewer 2 |
|--------|------------|------------|
| D1     | 1          | 1          |
| D2     | 1          | 0          |
| D3     | 0          | 1          |
| D4     | 1          | 0          |
| D5     | 1          | 1          |
| D6     | 1          | 0          |
| D7     | 0          | 1          |
| D8     | 1          | 0          |
| D9     | 1          | 1          |
| D10    | 1          | 0          |

Table 1. Data from inspection: '0' represents 'not found' and '1' represents 'found'.

Use the Lincoln-Peterson model to estimate the number of remaining defects in the document.

(1 mark)

Task 7 (3 marks + 3 bonus marks):

There exist several types of system tests, for example

- functional testing,
- performance testing,
- stress testing,
- configuration testing,
- security testing,
- recovery testing,
- usability testing,
- reliability testing.

Pick three types of system tests from the list above. Then briefly characterise each of the picked test approaches (in one or two sentences each) and give for each picked test approach a concrete example of what would be subject of testing.

(3 marks + 3 bonus marks)