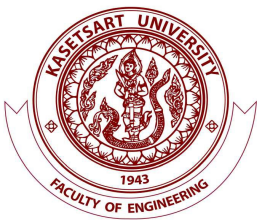



# Faculty of Engineering

# Kasetsart University

Final examination, 2013 - 2014, 2nd Semester
219343: Software Testing
Tuesday, 18th of March, 2014

Section 450
Lecturer: Uwe Gühl
1.00 pm until 3.00 pm

Name:_____ ID:_____

Points:_____/ 60 Mark:_____

Instructions: Read the following guidelines thoroughly before starting working on the exam

**1 Nature of exam and scoring rules**

1.1 There are Multiple-Choice-Questions and open questions on next pages. Maximum 60 points are available. You should attempt to complete every question. The exam spans 14 pages in total, including this cover page.

1.2 For the Multiple-Choice-Questions: Choose only ONE choice that you believe is correct (or most suitable). Mark your selected choice with a crossing sign (X) for each respective question. Marking more than one choice for each question is considered invalid and no points will be given. Exception: For question 4.4 select for each statement a crossing sign (X) in the correct column.

1.3 For the open questions: Enter the correct answer.

1.4 The points to be achieved are listed in every question.

**2 Exam policy during exam session**

2.1 No books, lecture notes, or any kind of documents, including calculators, are permitted.

2.2 Use only blue or black inked pens to write your name, student ID, on the exam sheet.

2.3 Do NOT separate any exam page from the exam set, or it will be considered an attempt to cheat.

2.4 Turn off all communication devices, or it will be considered an attempt to cheat.

2.5 No discussions/talking among students are permitted, or students involved will be considered cheating.

2.6 At the expiration of exam time, students are to return the complete exam set to the exam proctor.

**3 Policy for cheating**

Should you be caught for an attempt to cheat, regardless of the situations, you will automatically be given an F grade for this course, and be reported to the board of exam committees for further necessary disciplined penalties by the dean and chancellor offices.

| **Board of Examination Committee** | |
|---|---|
| Thanya Kiatiwat | (Head of Committee Members) |
| Putchong Uthayopas | (Committee Member) |
| Arnon Rungsawang | (Committee Member) |
| Uwe Gühl | (Committee Member) |

Name:_____ID:_____

# 1 Fundamentals of Testing [ / 6]

## 1.1 Which statement concerning early test design is NOT true? [ / 1]

a) Early test design might cause changes of requirements ☐
b) Faults detected during early test design are more expensive to fix ☒
c) Early test design can prevent fault multiplication ☐
d) Early test design can find faults ☐

## 1.2 In which order tests should be executed? [ / 1]

a) The most difficult should be tested first, so there is enough time for fixing ☐
b) The easiest tests should be executed first to achieve early confidence ☐
c) The most important test cases should be executed first ☒
d) The first tests prepared should be executed first as well ☐

## 1.3 What is the reason to divide testing into distinct stages? [ / 1]

a) It is easier to manage testing in stages ☐
b) It makes it possible to run different tests in different environments ☐
c) The more stages defined, the better the testing ☐
d) Each test stage has a different purpose ☒

## 1.4 Which of the following is NOT part of the Fundamental Test Process? [ / 1]

a) Test planning and control ☐
b) Requirement analysis ☒
c) Test implementation and execution ☐
d) Evaluating exit criteria and reporting ☐

## 1.5 The highest level of independence for verification and validation could be achieved by [ / 1]

a) a test organization outside the project's sphere of influence ☒
b) a test organization inside the project team ☐
c) test engineers ☐
d) developers assigned to testing tasks like unit testing and reviews ☐

## 1.6 What describes best the pesticide paradox testing principle? [ / 1]

a) Running the same tests over a period of time they loose their effectiveness to find new defects ☒
b) The poisonous atmosphere in case of communication issues between testers and developers ☐
c) If there are still severe defects in the software even if exhaustive testing was done ☐
d) A small number of modules is responsible for most of the operational failures ☐

# 2 Testing Throughout the Software Life Cycle [ / 5]

## 2.1 Which of the following requirements is testable? [ / 1]

a) The system shall be user friendly ☐
b) The safety-critical parts of the system shall contain zero faults (0 faults) ☐
c) The response time shall be less than one second for the specified design load ☒
d) The system shall be built to be portable ☐

### 2.2 In which test level you focus on possible defects in the interfaces and in the interaction between integrated components? [ / 1]

a) Acceptance Testing ☐
b) System Testing ☐
c) Integration Testing ☒
d) Component / Unit Testing ☐

### 2.3 To which test level belongs contract and regulation testing? [ / 1]

a) Acceptance testing ☒
b) System testing ☐
c) Integration testing ☐
d) Component / Unit Testing ☐

### 2.4 Smoke testing mainly helps in what? [ / 1]

a) To make sure that documentation like release notes are delivered ☐
b) To ensure that most crucial functions of a program work ☒
c) To assure that possible power cut issues are covered ☐
d) To save effort in retesting defects immediately ☐

### 2.5 Which statement is related to maintenance testing? [ / 1]

a) Final planning activity before starting test execution ☐
b) Synonym for testing the quality of service ☐
c) Triggered by modifications, migration or retirement of existing software ☒
d) Testing the level of maintenance by the vendor ☐

## *3 Static Techniques [ / 7]*

### 3.1 Which statement is TRUE? [ / 1]

a) Only failures could be detected during static testing ☐
b) Static testing starts early in the software life cycle and so it is done during the verification process ☒
c) Static testing requires normally the execution of a program, only few defined activities could be done without execution of the program ☐
d) Every static testing activity requires corresponding tools ☐

### 3.2 What are defined roles in reviews? [ / 1]

a) Test manager, tester, reviewers, author, optional moderator ☐
b) Author, reviewers, project manager, integration manager, test manager ☐
c) Scribe / recorder, team assistant, optional author, reviewers ☐
d) Manager, author, moderator, reviewers, optional scribe / recorder ☒

### 3.3 What of the following is part of the planning phase of a formal review? [ / 1]

a) Selecting the personnel and allocating roles ☒
b) Explaining the objectives ☐
c) Follow up, gathering metrics ☐
d) Individual meeting preparations ☐

### 3.4 Who typically uses static analysis tools? [ / 1]

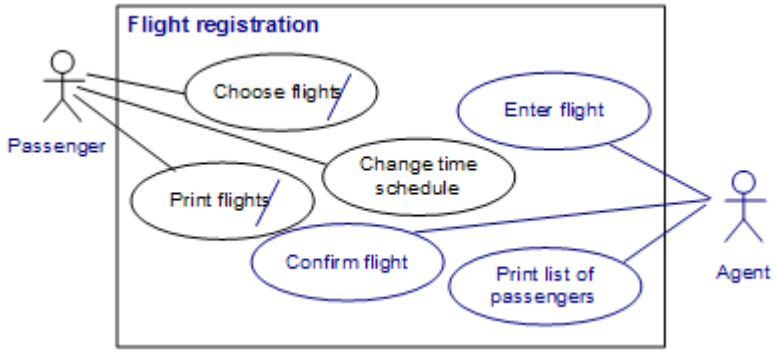
a) Test manager ☐
b) Tester ☐
c) Developers ☒
d) Tester and developers ☐

Name:_____ID:_____

### 3.5 Review of requirements [ / 3]

Following requirement for a flight registration program could be found in a specification:

1. Every passenger may choose a flight.
   It is possible to change the time schedule of a flight fast and easily. A passenger could print out flight information.
2. Agents could enter flights. An agent could confirm flights. An agent could print a list of passengers of a specific flight.




You are asked for a review. What would you update or change?
Which questions would you like to ask so that you could test given requirements?
Please add for each question a reasonable assumption as proposal.

- What means “fast” and “easily”?
  Proposal: The time schedule of a flight could be changed in every user interaction window, where the flight is visible. The change of the time schedule must be possible in 5 seconds.
- Use case diagram is incomplete
  For nouns singular form should be used instead of plural form to prevent misunderstandings.

## *4 Test Design Techniques* *[ / 22]*

### 4.1 What is a test design technique? [ / 1]

a) It is a process to measure in a test plan the testing activities to be done ☐
b) It is a way to measure the quality of software ☐
c) It is a process for selecting test cases ☒
d) It is a way to determine expected outputs ☐

### 4.2 Which kind of defects are typically detected using use case testing? [ / 1]

a) Defects in the interface parameters in integration testing ☐
b) Defects in the system as it transitions between one state and another ☐
c) Defects in the process flows during real-world use of the system ☒
d) Performance issues ☐

Name:_____ID:_____

## 4.3 Use case testing [ / 4]

You got following requirement for a file sharing website mm11.co.th:

It should be possible to upload files with a size of 1 MB or smaller. Altogether 20 MB webspace is available.
Of course it is possible to download these files as well.
It is possible that the owner of a file gives access rights to specific users (read only or read and write).
Supported are the browsers Skahari and Freenut, where Freenut is recommended.

a. How many test cases should be created? Give a brief explanation. [ / 2]

**20** test cases should be created.
– Main test cases:
Pre condition: All test cases to be executed on Freenut
* F1 Upload file with size 1 MB
* F2 Download file with size 1 MB
* F3 Reading access for friend on file is working
* F4 Writing access for friend on file is working
– Failure test cases
Pre condition: All test cases to be executed on Freenut
* FE1 Upload of a file with size > 1 MB not possible
* FE2 Upload of more files, if webspace has reached 20 MB, is not possible
* FE3_Upload of file with size 1 MB interrupted
* FE4 File with read access not reachable by other users
* FE5 File with read access could not be edited by friend
* FE6 File with write access not reachable by other users
– Alternative test cases
All test cases mentioned above to be executed with Skahari as well
S1 to S4, SE1 to SE6

b. Describe one test case in detail, following the template below: [ / 2]

| | |
|---|---|
| **Test Case Name** | Upload file with size 1 MB |
| **Test Case Id** | F1 |
| **Test Case Priority** | 1=high |
| **Pre conditions** | Browser freenut, user has account at mm11.co.th, webspace used ≤ 19 MB |
| **Post conditions** | used webspace increased:<br>old used webspace + 1MB = new used webspace |

| | | |
|---|---|---|
| **Description** | Successful upload file with size 1 MB | |
| **Role** | Owner | |
| **Steps** | **Step description** | **Expected result** |
| 10 | Opening website mm11.co.th | Web site opens with login page |
| 20 | Login with user "user", password "123" | Web site opens with file structure |
| 30 | Press "Upload" Button | File choosing windows open |
| 40 | Choose file "size1MB.mpeg" | Upload starts, status line info:<br>Upload of size1B.mpeg done |
| 50 | Logout | Logout page presented |

## 4.4 Decide, if following test design methods are either white box methods or black box methods. [ / 2]

| Test design method | White box | Black box |
|---|---|---|
| Decision coverage | x | |
| Boundary value analysis | | x |
| Equivalence partitioning | | x |
| Cyclomatic complexity | x | |
| Code inspection | x | |
| Decision tables | | x |
| State transition testing | | x |
| Use case testing | | x |

## 4.5 Abstract / Concrete test cases [ / 1]

Explain the difference between abstract test cases and concrete test cases.

**Abstract (logical, high level) test case**
A test case without concrete (implementation level) values for input data and expected results. Logical operators are used; instances of the actual values are not yet defined and/or available

**Concrete (low level) test cases**
A test case with concrete (implementation level) values for input data and expected results. Logical operators from high level test cases are replaced by actual values.

## 4.6 Describe the Test-driven development cycle. [ / 1]

- Add a test
  Each new feature begins with a test based on requirements.
- Run all tests and check, if the new test fails
  The added test should fail.
- Write some code
  New code should be added, so that the code meets all tested requirements.
- Run all tests
  Now all tests should pass.
- Refactor code
  If necessary, code can be cleaned up.

- Repeat
  Now new requirements could be implemented by starting the cycle again.

## 4.7 How is McCabe's cyclomatic complexity M defined? [ / 1]

If L = the number of edges/links in a graph
N = the number of nodes in a graph
P = the number of disconnected parts of the graph (e.g. a called graph or subroutine)

a) M = N – L + 2P ☐
b) M = L – N + 2P ☒
c) M = N + L – 2P ☐
d) M = L + N + 2P ☐

## 4.8 Code coverage and code complexity [ / 1]

You got following report from a code coverage tool concerning a Java project:

| | |
|---|---|
| Statement coverage: | 100 % |
| Decision coverage: | 55 % |
| Code complexity: | 38 |

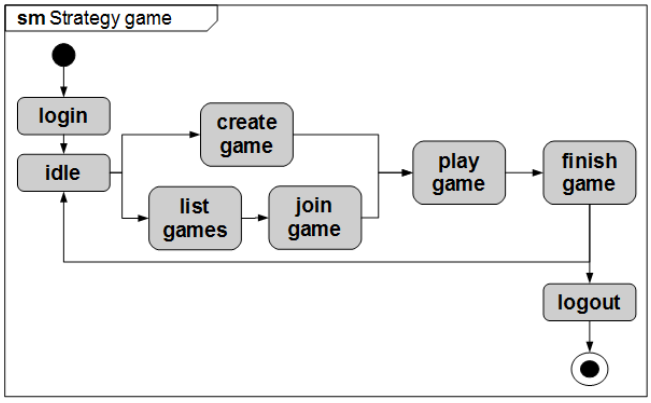
What should be the next activities to increase the quality?

- Increasing the number of test cases
  100% statement coverage is achieved, but this is a really weak criteria.
  100 % decision coverage should be considered as a minimal requirement.
  There should be an explanation by the responsible developer, if and why 100 % decision coverage is not achieved.

- Refactoring the code to reduce the code complexity
  It has been shown, that a code complexity of 38 had a probability of 50% of being fault-prone.

## 4.9 State transitions [ / 1]

Given following state transition diagram:




Which of the following series of state transitions contains an INVALID transition?

a) login, idle, create game, play game, finish game, logout ☐
b) login, idle, create game, join game, play game, finish game, logout ☒
c) login, idle, list games, join game, play game, finish game, logout ☐

d) login, idle, list games, join game, play game, finish game, idle, create game, play game, finish game, logout ☐

## 4.10 JUnit Test Cases [ / 6]

A system was designed to return a month name by a given number.
Following code is available:

```
public class monthCounter {
   String monthsName (int number) {
        switch(number) {
            case  1: return "January";
            case  2: return "February";
            case  3: return "March";
            case  4: return "April";
            case  5: return "May";
            case  6: return "June";
            case  7: return "July";
            case  8: return "August";
            case  9: return "September";
            case 10: return "October";
            case 11: return "November";
            case 12: return "December";
            default: throw new IllegalArgumentException(
                    "Unknown month number: " + number );
        }
   }
}
```

Following JUnit test cases have been prepared so far:

```
public class monthCounterTest {

   @Test(expected=IllegalArgumentException.class)
   public void testIllegalArgument0() throws Exception {
      monthCounter tester = new monthCounter();
      tester.monthsName(0);
   }

   @Test
   public void testMonthsName1() {
      monthCounter tester = new monthCounter();
      assertEquals("Result", "January", tester.monthsName(1));
   }

   @Test
   public void testMonthsName12() {
      monthCounter tester = new monthCounter();
      assertEquals("Result", "December", tester.monthsName(12));
   }

   @Test(expected=IllegalArgumentException.class)
   public void testIllegalArgument13() throws Exception {
      monthCounter tester = new monthCounter();
      tester.monthsName(13);

   }
```

```
}
```

a. Concerning the presented source code: What is the value of McCabe's Cyclomatic Complexity M? [ / 1]

M = L – N + 2 P = 26 – 15 + 2 x 1 = 13

b. Concerning the given test cases: How much is the decision coverage of the presented source code? [ / 1]

Tested decisions / all decisions = 3/13 = 23.1 %

c. How many test cases should be added if you follow a defined test design technique? Give a brief explanation. [ / 1]

**1**, if we follow equivalence partitioning (3 test cases required – Invalid partition 0, valid partition 1; invalid partition from 13 up is missing)
**1**, if we test boundary values (4 test cases required – 0, 1, 12; 13 is missing)

d. Add one test case in the test code above. [ / 1]

After updating the code, following refactored source code is available:

```
public class MonthCounter {

   String monthsName (int number) {
      ArrayList<String> monthList = new ArrayList<String>();
            monthList.add("January");     monthList.add("February");
            monthList.add("March");       monthList.add("April");
            monthList.add("May");         monthList.add("June");
            monthList.add("July");        monthList.add("August");
            monthList.add("September");   monthList.add("October");
            monthList.add("November");    monthList.add("December");

      if (number <= (monthList.size()) && number > 0)
         return monthList.get(number-1);
      else
         throw new IllegalArgumentException("Unknown month number");
   }
}
```

e. Concerning the refactored source code:
What is now the value of McCabe's Cyclomatic Complexity M? [ / 1]

M = L – N + 2 P = 6 – 5 + 2 x 1 = 3 M = b + 1 = 2 + 1 = 3

f. If you take the prepared JUnit test cases including your added JUnit test case.
What is now the decision coverage of the refactored source code? [ / 1]

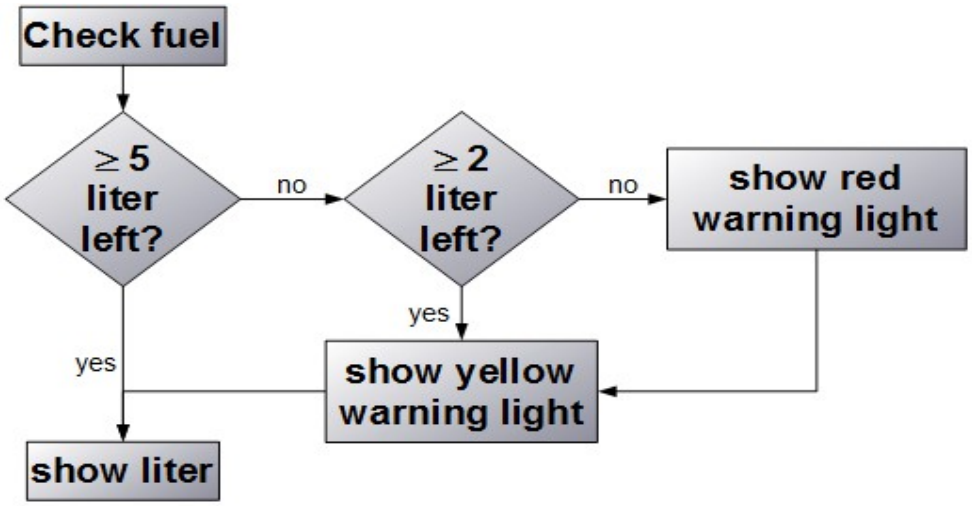
Tested decisions / all decisions = 3/3 = 100 %

## 4.11 Which of the following could be a coverage measure for state transition testing? [ / 1]

a) The response time for each transaction is adequate ☐
b) All boundaries have been exercised ☐
c) Combinations of inputs that might not otherwise have been exercised ☐
d) Every transition has been exercised ☒

## 4.12 Statement coverage and decision coverage [ / 2]

Following graph is representing a program displaying the liter of fuel in a tank, and additionally to show warning lights if required.




Requested are test cases representing liter of fuel in the tank.

a. How many tests are necessary to achieve 100 % statement coverage? [ / 0.5]

**1**

b. Write down all the required tests to achieve 100 % statement coverage. [ / 0.5]

1 liter

c. How many tests are necessary to achieve 100 % decision coverage? [ / 0.5]

**3**

d. Write down all the required tests to achieve 100 % decision coverage. [ / 0.5]

5 liter, 3 liter, 1 liter

Name:_____ID:_____

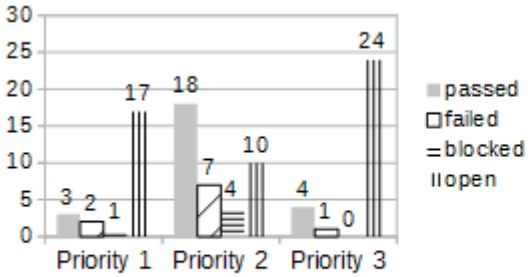
# 5 Test Management [ / 16]

## 5.1 Which of the following a test manager should NOT do? [ / 1]

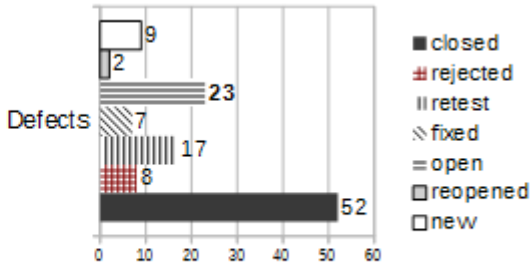
a) Provide information for risk analysis and quality improvement ☐
b) Re-allocate resources to meet original plans ☒
c) Decide what should be automated, to what degree, and how ☐
d) Sign the system off for release ☐

## 5.2 Test report [ / 3]

Find below a test report.




Test Case Overview




Defect Overview

a. What does it mean, that test cases are in status “blocked”? [ / 1]

Blocked test cases could not be executed, if e.g. they depend on the execution of failed test cases as pre condition or if the corresponding functionality is not available yet.

b. Which information is missing concerning the defect overview? [ / 1]

Severity of defects.

c. As a responsible test manager, what would be your request to the test team based on given information out of the test report? [ / 1]

– 17 open Priority 1 test cases should be executed as soon as possible.
– 17 defects in the status “ready for retest” should be retested as soon as possible.

## 5.3 What describes best the test monitoring? [ / 1]

a) Measures of tracking process ☒
b) Effort required to perform activities ☐
c) Re-allocation of resources when test overruns ☐
d) Calculation of required test resources. ☐

## 5.4 What test items should be put under configuration management? [ / 1]

a) The test object, the test material, and the test environment ☒
b) The problem reports and the test material ☐
c) Only the test object. The test cases need to be adapted during agile testing ☐
d) The test object and the test material.
Not the test environment as this is not related to the final software delivery ☐

Name:_____ID:_____

## 5.5 Explain the meaning of “severity” and “priority” in incident management! [ / 2]

- Severity
  How severe the impact on the system is.
  Typically values start with 1 (= very high, like system crash, data loss) to n (=very low, like spelling mistakes).
- Priority
  How urgent a defect has to be fixed.
  Typically values start with 1 (= very urgent, like a hot fix deployment) to n (=very low, like if time is left).
  Possible special status: “must not be fixed”.

## 5.6 Incident management [ / 3]

You are a defect manager in a software development team with test manager Matt, test engineer Ted, developer Danny, and customer Carl. Given following open defect overview from today.

| ID | Summary | Status | Assigned To | Detected By | Severity | Detected on Date | Last Modified on Date | Test Case Reference |
|---|---|---|---|---|---|---|---|---|
| 23 | Portal: Cookies without HttpOnly fla | Open | Danny | Ted | 1 - Critical | 06.02.14 | 07.02.14 | TC-603 |
| 47 | Portal: Corrupt XSS in upload functi | Open | Danny | Carl | 1 - Critical | 17.12.13 | 11.01.14 | |
| 496 | Portal: No error message pops up | New | Matt | Ted | 1 - Critical | 06.02.14 | 06.02.14 | TC-007 |
| 38 | Apache 2.2 < 2.2.24 Multiple Cross | Ready for Retest | Ted | Ted | 2 - Major | 12.03.14 | 17.03.14 | TC-264 |
| 376 | Portal: Reassign Admin right to an | Fixed | Danny | Carl | 2 - Major | 12.03.14 | 14.03.14 | TC-289 |
| 446 | Outdated PHP for remote web serve | Ready for Retest | Ted | Ted | 2 - Major | 13.03.14 | 17.03.14 | |
| 447 | Unencrypted Server | Ready for Retest | Ted | Ted | 2 - Major | 17.12.13 | 03.02.14 | |
| 448 | Apache HTTP Server to be restarte | Open | Danny | Ted | 3 - Minor | 18.12.13 | 14.03.14 | TC-144 |
| 449 | Test case TC-202 wrong | Open | Ted | Carl | 3 - Minor | 17.12.13 | 17.12.13 | TC-202 |
| 129 | Portal: Automated logout after 1 ho | Ready for Retest | Ted | Carl | 4 - Feature | 14.01.14 | 07.02.14 | |
| 130 | Portal: GUI color scheme to be upd | New | Matt | Matt | 4 - Feature | 16.03.14 | 16.03.14 | TC-412 |

a. Which important measures should be taken? [ / 2]

Important as defects have severity 1 or 2:

1. Defect 23: Clarification of status with Danny
   Reason: No updates since 07.02.14, although it's critical.
2. Defect 47: Clarification of status with Danny
   Reason: No updates since 11.01.14, although it's critical.
   Request to Ted to prepare a corresponding test case
3. Defect 496: Request to Matt to decide about the defect, either to reject it or to assign it to a project team member.
4. Defect 446: Request to Ted to prepare a corresponding test case.
5. Defect 447: Request to Ted to prepare a corresponding test case and to retest the defect, no updates since 03.02.14.

b. Which additional measures should be recommended? [ / 1]

Additional measures as defects have severity 3 or 4:

1. Defect 449: Request to Ted to update the test case, no updates since 17.12.13.
2. Defect 129: Request to Ted to retest the defect, no updates since 07.02.14, and to prepare a corresponding test case

Name:_____ID:_____

## 5.7 Which information is missing in the following incident report ? [ / 1]

| | |
|---|---|
| Title: | Playing a movie using a streaming service, it stops after 2 minutes. It happens every time. |
| Expected result: | Playing of the movie should continue until the end. |
| Severity: | High |
| Priority: | Urgent |

a) Assignment, who should proceed working on the incident ☐
b) Id of the incident ☐
c) Actual result ☐
d) Identification (Software and hardware) of the streaming service ☒

## 5.8 Review an incident report [ / 2]

| Attribute | Values |
|---|---|
| Incident Number | 1 |
| Summary | Update of ruler position in slide master via mouse click not possible |
| Detected on date | 15.03.2014 |
| ... by author | |
| Assigned to | |
| | |
| Product | NiceOffice |
| Detected in version | 4.2 |
| Severity | |
| Priority | 2=Urgent |
| Status | New |
| Description | Description: Update of ruler position to change the indention in the slide master with left-hand mouse click not possible.<br><br>Steps to Reproduce:<br>Open an empty presentation .<br>Open the master (Menu View → Master → Slide Master).<br>In the ruler do left-hand mouse click to move the arrow to change the indention.<br>Update and leave mouse click.<br><br>Expected Results: Ruler position gets updated<br><br>Actual Results: Hour glass appears, only closing application with data loss possible |
| Attachments | Master-Ruler-Crash.gif |
| Comments | Operating System: Windows7 |
| Links | Testcase 114 |

a. Which most important information is missing? [ / 1]

Author of defect

b. What else are review findings? [ / 1]

1. Severity should be given
2. “Assigned to” should be determined
3. Steps should be numbered.

### 5.9 Poor software characteristics are which kind of risk? [ / 1]

a) Project risk ☐
b) Product risk ☒
c) Project risk and product risk ☐
d) It is neither a project risk nor a product risk as it is a basic assumption ☐

### 5.10 Which of the following statement is TRUE? [ / 1]

a) An incident may be closed without being fixed ☒
b) An incident record should not include information on test environments ☐
c) The final stage of incident tracking is ready for retest ☐
d) Incidents may not be raised against test cases ☐

## *6 Tool support for testing [ / 4]*

### 6.1 With which of the following categories a test comparator tool is in general associated? [ / 1]

a) Tool support for performance and monitoring ☐
b) Tool support for static testing ☐
c) Tool support for the management of testing and tests ☐
d) Tool support for test execution and logging ☒

### 6.2 In which of the following activities the biggest potential cost saving is possible in introducing tool support for testing? [ / 1]

a) Test management ☐
b) Test planning ☐
c) Test design ☐
d) Test execution ☒

### 6.3 How to start if you want to introduce a new testing tool? [ / 1]

a) Attend a tool exhibition ☐
b) Invite a vendor to give a demonstration ☐
c) Analyse your needs and requirements ☒
d) Find out what your budget would be for the tool ☐

### 6.4 Which statement regarding the success factors for the effective use of a test tool is TRUE? [ / 1]

a) The implementation of a new tool should follow a “Big Bang” approach ☐
b) The test process should be defined in parallel with a tool deployment ☐
c) It is important to provide training and coaching for new users ☒
d) Users should not get confused with monitoring the tool use ☐