Faculty of Engineering Kasetsart University

Final Examination, 2nd Semester, 2015 / 2016
01219343 Software Testing and Software Project Management
Thursday, 24th of March, 2016

Section 450
Lecturer: Uwe Gühl
9.00 am until 12.00 am

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 17 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 problem. Marking more than one choice for each problem is considered invalid and no points will be given. Exception: For questions 2.5, 4.6 and 5.7 select for each topic 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

Peerayuth Charnsethikul	(Head of Committee Members)
Anan Phonphoem	(Committee Member)
Arnon Rungsawang	(Committee Member)
Uwe Gühl	(Committee Member)

Name:____ID:__________________

1 Fundamentals of Testing [/ 10]

1.1 Which of the following is NOT a testing objective? [/ 1]

a) Finding defects. □
b) Gaining confidence about the level of quality and providing information. □
c) Preventing defects. □
d) Debugging defects. ☒

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 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. □

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

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

1.5 Which statement concerning early testing is TRUE? [/ 1]

a) Early test design might cause changes of requirements. ☒
b) Faults found during early test design are more expensive to fix. □
c) Early test design cannot prevent fault multiplication. □
d) Early test design takes more effort than late test design. □

1.6 How much testing is enough? [/ 1]

a) When all testing has been done as planned. ☐
b) It's depending on the size of the testing team. ☐
c) It's depending on the level of risk. ☒
d) When it could be proved that the system works correctly. ☐

1.7 As part of which process activity do you determine the exit criteria? [/ 1]

a) Test planning and control. ☒
b) Evaluating exit criteria and reporting. ☐
c) Test closure activities. ☐
d) Test implementation and execution. ☐

1.8 Why should tests get prioritized? [/ 1]

a) to shorten the time required for testing. ☐
b) to do the best testing in the time available. ☒
c) to do more effective testing. ☐
d) to find more defects. ☐

1.9 Describe two possibilities to improve communication and relationships between testers and others [/ 2]

- Start with collaboration rather than battles.
  The common goal of everyone should be: Better quality systems
- Communicate findings on the product in a neutral, fact-focused way, e.g. reproducible defect descriptions
- Write objective and factual incident reports and review findings.
- Do not criticize the person who created it.
- Try to understand how the other person feels and why they react as they do.
- Confirm that the other person has understood what you have said and vice versa.

Name:____ID:__________________

2 Testing Throughout the Software Life Cycle [/ 6]

2.1 Why is a bidirectional traceability between test conditions and requirements requested? [/ 1]

a) for better calculation of the test effort in test execution phase. □
b) to better assign testers to corresponding test cases. □
c) for impact analysis when requirements change. ☒
d) to optimize defect reports. □

2.2 Which statement is TRUE? [/ 1]

a) Re-testing looks for unexpected side effects; regression testing is repeating those tests. □
b) Re-testing is done by developers, regression testing is done by independent testers. □
c) Re-testing means running a test again after a fix; regression testing looks for unexpected side effects. ☒
d) Re-testing is done after faults are fixed; regression testing is done earlier. □

2.3 What is alpha testing? [/ 1]

a) Pre-release testing by end user representatives at their site. □
b) Pre-release testing by end user representatives at the developer's site. ☒
c) Post-release testing by end user representatives at their sites. □
d) Post-release testing by end user representatives at the developer's site. □

2.4 What is maintenance testing? [/ 1]

a) Updating tests when a system has changed. □
b) Acceptance testing by users to ensure that the system fulfils requirements. □
c) Retesting of defects during the development phase. □
d) Testing a released system that has changed. ☒

2.5 Decide, if following requirements are functional or non-functional requirements! [/ 2]

Requirement	Functional	Non-functional
The system displays the number of books and novels in the database.	x	
A user creates a request. It includes selecting start location and destination on map, recipient, dimension and weight of parcel, shipment end date and time, and price.	x	
The current position of the parcel should be displayed on the map in less than 10 seconds.		x
The command “next” will load another novel into the web site.	x	
Up to 10,000 users public user should be able to use the application in parallel.		x
It must be possible to use Robot World on Internet Explorer Version 11.0.28 and higher, Firefox 45.0.1 and higher, and Google Chrome 49.0.2623.87 and higher.		x

3 Static Techniques [/ 7]

3.1 With static analysis it is NOT possible to find [/ 1]

a) the use of a variable before it is defined. □
b) memory leaks. ☒
c) array bound violations. □
d) unreachable or dead code. □

3.2 Which of the following are the main phases of a formal review? [/ 1]

a) Initiation, status, preparation, review meeting, rework, follow up. □
b) Planning, preparation, review meeting, rework, closure, follow up. □
c) Planning, kick off, individual preparation, review meeting, rework, follow up. ☒
d) Preparation, review meeting, rework, closure, follow up, root cause. □

3.3 Could reviews or inspections be considered part of testing?[/ 1]

a) Yes, because both help detect faults and improve quality. ☒
b) Yes, because testing includes all non-constructive activities. □
c) No, because they apply to development documentation. □
d) No, because they are normally applied before testing. □

3.4 Advantages of reviews [/ 1]

List at least three advantages of reviews:

- Early defect detection and correction
- Development productivity improvements
- Reduced development timescales
- Reduced testing cost and time
- Lifetime cost reductions
- Fewer defects and improved communication.
- Reviews can find omissions for example, in requirements, which are unlikely to be found in dynamic testing.

3.5 Which tool should be used to detect unreachable code? [/ 1]

a) A static analysis tool. ☒
b) A dynamic analysis tool. □
c) A coverage tool. □
d) Complexity calculation tool. □

3.6 Data flow analysis [/ 1]

Describe possible anomalies in data flow analysis.

- dd (defined / defined)
  Defined, then gets defined again before first value gets used
- du (defined / undefined)
  Defined, then gets invalid or undefined without use
- ur (undefined / referenced)
  Undefined variable read or used

3.7 Data flow analysis [/ 1]

What is the result of a data flow analysis of following code?

```
1  public class Foo {
2    public int foo() {
3      int leapyear = 2559;
4      leapyear = 2016;
5      return leapyear;
6      }
7  }
```

- dd anomaly with leapyear
  leapyear is defined in line 3, and again in line 4 without being used.

Name:____ID:____________________

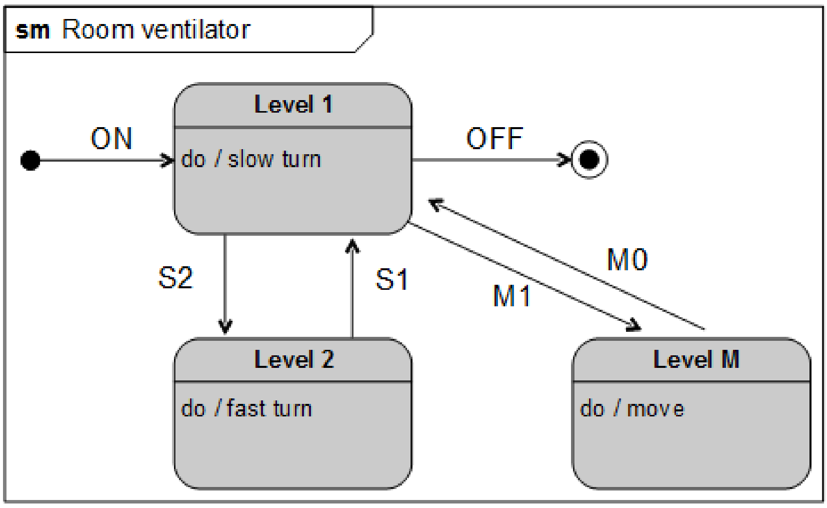
4 Test Design Techniques [/ 20]

4.1 At which test levels functional testing may be performed? [/ 1]

a) At system and acceptance testing levels only. □
b) At all test levels. ☒
c) At all test levels, but not at integration testing. □
d) At the acceptance testing level only. □

4.2 State diagram [/ 2]

The following state diagram is given:



i. Test Coverage [/ 1]
Which of the test cases below will cover the following series of state transitions?
Level 1 – Level M – Level 1 – Level 2 – Level 1 – Final state
a) ON, M1, M0, S2, OFF □
b) M1, M0, S1, S2, S1, OFF □
c) ON, M1, M0, S2, S1, OFF ☒
d) ON, M1, M0, S2, OFF □

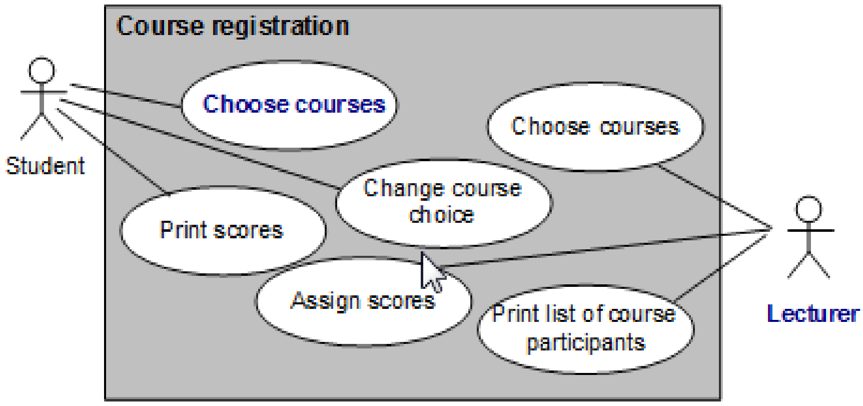
ii. Invalid state transition [/ 1]
Which of the following represents an INVALID state transition?
a) Initial state to Level 1 □
b) Level 2 to Level 1 □
c) Level 1 to Level 2 □
d) Level M to Final state ☒

4.3 Use Case Diagram [/ 1]

You got following requirements description:

1. The system “Course registration” should offer for students the possibility to choose a course to attend, to change their choice in case, and to print out their scores.
2. The system should offer for lecturers that they could choose courses, assign scores to students, and to print the list of participants of a specified course.

Enter the missing information in the Use Case Diagram below.



4.4 Code coverage and code complexity [/ 2]

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

Statement coverage:	100 %
Decision coverage:	34.8 %
Code complexity:	40

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

4.5 Use case testing [/ 3]

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

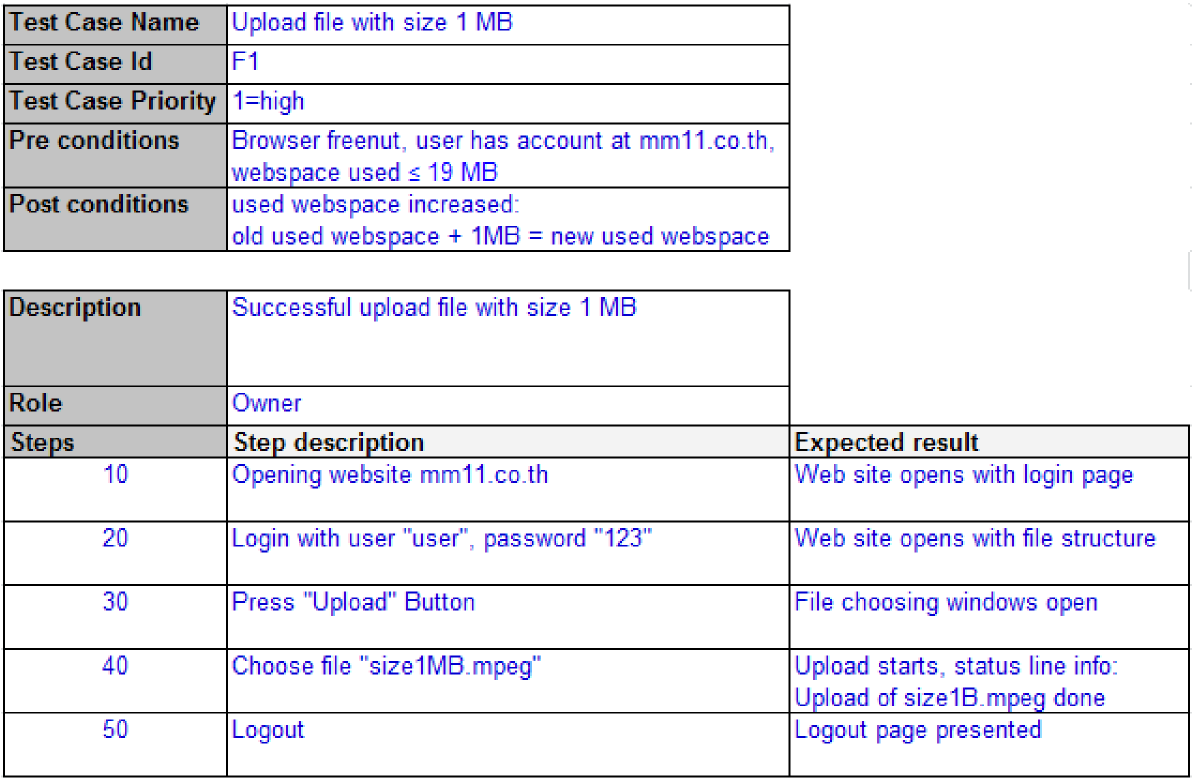
It should be possible to upload files with a size of 1 MB or smaller. Altogether 20 MB webspace is available.

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.

Design one test case in detail, following the template below:



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: old used webspace + 1MB = new used webspace

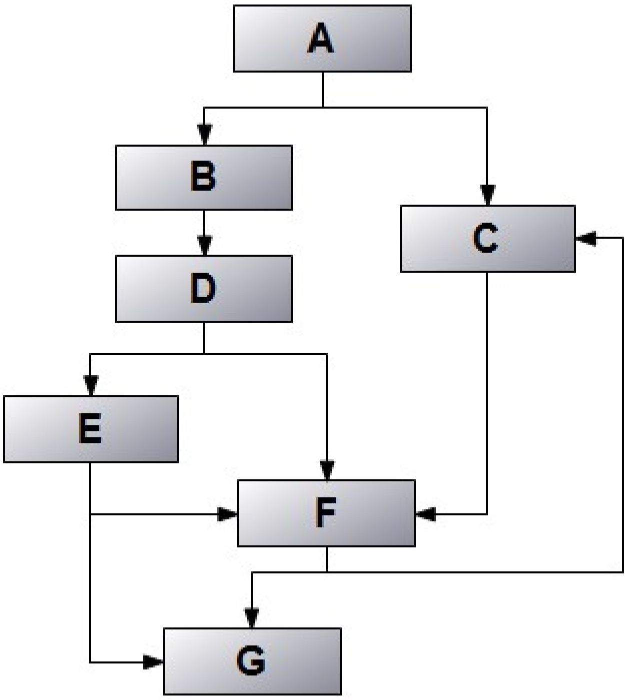
Description	Successful upload file with size 1 MB	
Role	Owner	
Steps	Step description	Expected result
10	Opening website mm11.co.th	Web site opens with login page
20	Login with user "user", password "123"	Web site opens with file structure
30	Press "Upload" Button	File choosing windows open
40	Choose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

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

Test design method	White box	Black box
Boundary value analysis		x
Code inspection	x	
Decision coverage	x	
Decision tables		x
Design inspection	x	
Error guessing		x
State transition testing		x
Use case testing		x

4.7 Coverage and Code Complexity [/ 4]

Following graph is representing a program:



a. How many tests are necessary to achieve 100 % Statement Coverage? Write down, what paths the required tests should cover. [/ 1]

1

A, B, D, E, F, C, F, G

b. How many tests are necessary to achieve 100 % Decision Coverage? Write down, what paths the required tests should cover. [/ 2]

4

A, B, D, E, G; A, B, D, E, F, G;

A, B, D, F, G; A, C, F, C, F, G.

c. What is the value of McCabe's Cyclomatic Complexity M? [/ 1]

M = L – N + 2 P = 10 – 7 + 2 x 1 = **5,** or **M =** b + 1 = 4 + 1 = **5 (binary conditions)**

4.8 JUnit Test Cases [/ 4]

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

```
public class dayCounter {
   String dayName (int number) {
        switch(number) {
         case  1: return "Monday";
         case  2: return "Tuesday";
         case  3: return "Wednesday";
         case  4: return "Thursday";
         case  5: return "Friday";
         case  6: return "Saturday";
         case  7: return "Sunday";
         default: throw new IllegalArgumentException(
                    "Unknown day number: " + number );
        }
   }
}
```

Following JUnit test cases have been prepared so far:

```
public class dayCounterTest {

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

   @Test
   public void testDayName1() {
      dayCounter tester = new dayCounter();
      assertEquals("Result", "Monday", tester.dayName(1));
   }

   @Test
   public void testDayName7() {
      dayCounter tester = new dayCounter();
      assertEquals("Result", "Sunday", tester.dayName(7));

   }

   @Test(expected=IllegalArgumentException.class)
   public void testIllegalArgument8() throws Exception {
      dayCounter tester = new dayCounter();
      tester.dayName(8);

   }

}
```

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

M = L – N + 2 P = 16 – 10 + 2 x 1 = 8

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

Tested decisions / all decisions = 2/8 = 25 %

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 8 up is missing)
- **2**, if we test boundary values (classical approach)
4 test cases required – 0, 1, 7, 8; missing: 7, 8)
- **6**, if we test boundary values (methodical approach)
(8 test cases required – -1, 0, 1, 2, 6, 7, 8, 9; missing: -1, 2, 6, 7, 8, 9)

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

4.9 Test-driven development [/ 1]

Describe the Test-driven development cycle.

- 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.

Name:____ID:__________________

5 Test Management [/ 12]

5.1 What of the following is least important for a Test Manager? [/ 1]

a) Configuration management. □
b) De-bugging. ☒
c) Estimating test duration. □
d) Incident Management. □

5.2 Should the test environment be under configuration management? [/ 1]

a) Yes, because a tester may need to restore the test environment. ☒
b) Yes, because the testers need this information to install the test object for running the test. □
c) No, because it is not part of the test object. □
d) No, because configuration management concerns only the test object and test material. □

5.3 Which of the following alternatives are typical tester tasks (as opposed to test manager)? [/ 1]

a) Set up configuration management of testware;
review tests developed by others. □
b) Plan and initiate the specification, preparation, implementation and execution of tests and monitor and control the execution. □
c) Decide what should be automated, to what degree, and how. □
d) Prepare and acquire test data; Review tests developed by others. ☒

5.4 How are “severity” and “priority” defined in incident management? [/ 1]

- 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”.

Name:____ID:____________________

5.5 Which of the following statements is TRUE? [/ 1]

a) An incident record should not include information on test environments. □
b) The final stage of incident tracking is fixing. □
c) Incidents may not be raised against documentation. □
d) An incident may be closed without being fixed. ☒

5.6 Which of the following does NOT help in monitoring the Test Progress? [/ 1]

a) Percentage of Test Case Execution. □
b) Percentage of work done in test environment preparation. □
c) Defect Information e.g. defect density, defects found and fixed. □
d) The size of the testing Team and skills of the engineers. ☒

5.7 Decide, if following risks are product risks or project risks. [/ 2]

Topic	Product risk	Project risk
Political problems and delays in especially complex areas in the product.		x
Poor software characteristics.	x	
Skill and staff shortages.		x
Possible reliability defect (bug).	x	
Error-prone areas, potential harm to the user, poor product characteristics.	x	
Failure-prone software delivered.	x	
Problems in defining the right requirements, potential failure areas in the software or system.		x
Low quality of requirements, design, code and tests.		x

Name:____ID:____________________

5.8 Incident report [/ 4]

Content of an incident report:

Tester Joe: I executed Test Case 2009 on 20.03.2016. First I opened an empty presentation using NiceOffice, version 5.1, operating system Windows 10, and entered some words. Then I opened the master (Menu View → Master → Slide Master). In the ruler I tried with left-hand mouse click to move the arrow to change the indention. When I have updated and released mouse click, an hour glass appeared, and I couldn't use NiceOffice anymore.

Only solution: Closing NiceOffice, but then all data are lost. Took a screenshot "Master-Ruler-Crash.gif". Found a work around: Changing ruler position via Menu is possible.

Write a structured incident report in the template below.

Attribute	Value
Incident Number	1
Summary	Update of ruler position in slide master via mouse click not possible
Detected on date	20.03.16
… by author	Joe
Assigned to	Test Manager
Product	NiceOffice
Detected in version	5.1
Severity	2=High
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. **Steps to Reproduce:** 1. Open an empty presentation. 2. Entering some words. 3. Open the master (Menu View → Master → Slide Master). 4. In the ruler do left-hand mouse click to move the arrow to change the indention. 5. Update and release mouse click. **Expected Results:** Ruler position gets updated. **Actual Results:** Hour glass appears, only closing application with data loss possible.
Attachments	Master-Ruler-Crash.gif"
Comments	Operating System: Windows 10 Work around possible: Changing ruler position via Menu.
Links	Test Case 2009

6 Tool Support for testing [/ 5]

6.1 How to start if a new testing tool is to be introduced? [/ 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. □

6.2 Which statement regarding the success factors for the deployment of a test tool is TRUE? [/ 1]

a) To succeed in the deployment of a new tool, it is important to provide training and coaching for new users. ☒
b) Defining a test process in parallel with a tool deployment is recommended, if you want to achieve success. □
c) Implementing the new tool in a Big Bang implementation will ensure success. □
d) To succeed in the deployment of a new tool, only hard work is needed. □

6.3 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. ☒

6.4 Which statement is TRUE? [/ 1]

a) Performance testing tools can’t be used if you are testing web applications. □
b) Performance testing tools can be used to simulate large numbers of users on the system under test. ☒
c) Performance testing tools are not able to simulate a load for more than 24 hours. □
d) Performance testing tools can be used to derive the complexity of the code. □

6.5 How is it called, if a test tool is intrusive? [/ 1]

a) Measurement errors. □
b) Fault attack. □
c) Probe effect. ☒
d) Intrusive factor. □