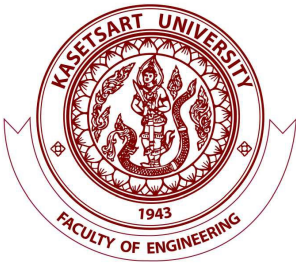


Faculty of Engineering

Kasetsart University

Midterm Examination, 2nd Semester, 2013 / 2014 — Section 450
219343 Software Testing — Lecturer: Uwe Gühl
Tuesday, 18th of February, 2014 — 2.00 pm until 4.00 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 problem. Marking more than one choice for each problem is considered invalid and no points will be given. Exception: For question 2.7 and 4.1 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)

1 Fundamentals of Testing [/ 17]

1.1 What is the goal of Software Tests? [/ 3]

- Finding defects
- High test coverage
- Statements concerning software quality

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

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

1.3 Why is testing necessary? [/ 1]

a) because testing is included in the project plan ☐
b) to fill the time between delivery of the software and the release date ☐
c) because software is likely to have faults ☒
d) to prove that the software has no faults ☐

1.4 When should testing be finished? [/ 1]

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

1.5 Which of the following is NOT an exit criteria? [/ 1]

a) Measures concerning code coverage, functionality or risk ☐
b) Verifying the test environment ☒
c) Remaining risk like lack of test coverage in defined areas or open defects ☐
d) Guesses of defect density or reliability measures ☐

1.6 Why should tests be prioritized? [/ 1]

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

1.7 How do you call something that is incorrect in software? [/ 1]

a) An error ☐
b) A failure ☐
c) A mistake ☐
d) A defect ☒

1.8 Which of the following is a benefit of test independence? [/ 1]

a) It does not require familiarity with the code ☐
b) It is less expensive than using developers to test their own code ☐
c) It avoids author bias in defining effective tests ☒
d) Testers are more efficient at finding defects than developers ☐

1.9 Which of the following is NOT an activity in the Fundamental Test Process? [/ 1]

a) Test Planning and Control ☐
b) Test Implementation and Execution ☐
c) Test Automation activities ☒
d) Evaluating Exit criteria and reporting ☐

1.10 What are the main activities during test planning? [/ 1]

a) To identify main requirements and risk areas to define objectives of testing ☒
b) To ensure that the test case specification is complete ☐
c) To determine the test design specification ☐
d) To know when a specific test has finished its execution ☐

1.11 Why is it typically not possible to do exhaustive testing, meaning to test all inputs and preconditions? [/ 1]

- Due to the combinatorial explosion of input parameter, complete testing is neither theoretically, nor practically possible.

1.12 What is the main reason to start testing early? [/ 2]

- To find defects early.
- As costs of defects increase, the later they are detected, early testing saves costs.

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

- Start with collaboration rather than battles. Common goal of everyone: 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.

2 Testing Throughout the Software Life Cycle [/ 13]

2.1 Which of the following is a Non-Functional test activity? [/ 1]

a) Unit testing ☐
b) Re-testing and Regression testing ☐
c) Smoke test ☐
d) Load and Performance testing ☒

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 characteristics could be used for the evaluation of software quality? [/ 1]

a) Price, margin, sales count ☐
b) Functionality, reliability, usability, efficiency, maintainability, portability ☒
c) Release notes, product description, user manual, installation guidelines ☐
d) White paper contents, customer presentation ☐

2.4 Which of the following is NOT a type of incremental testing approach? [/ 1]

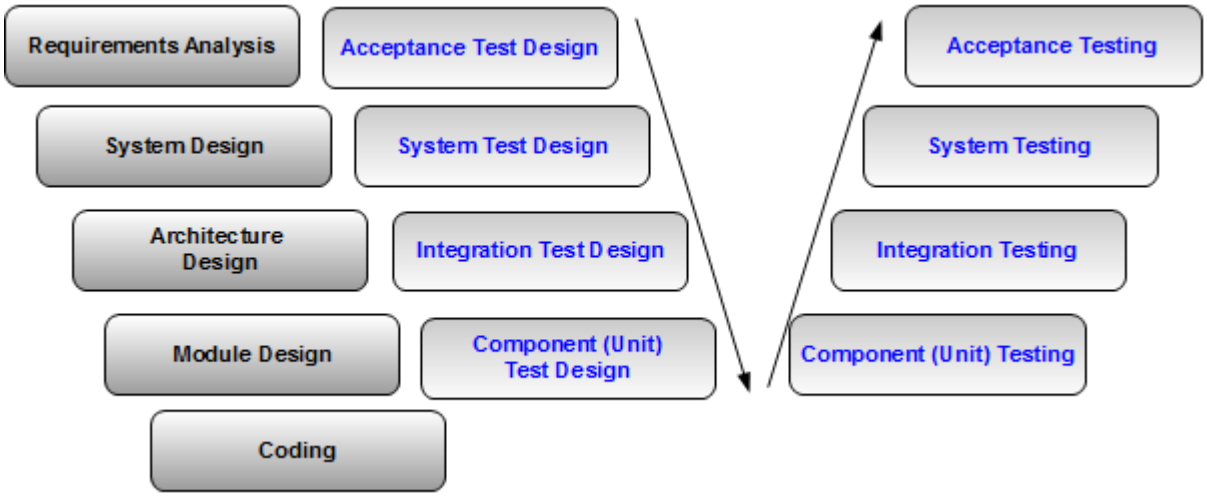
a) Alpha testing ☒
b) Functional incrementation ☐
c) Bottom up ☐
d) Top down ☐

2.5 Maintenance testing is [/ 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 ☒

2.6 'V-Model' following ISTQB [/ 4]

Add the testing and test preparation activities, integrated into each phase of the software development lifecycle according to the extended V-Model below following ISTQB.



2.7 Decide, if following requirements are functional or non-functional requirements! [/ 3]

Requirement	Functional	Non-functional
System is designed for IE Version 7 to 11, Mozilla Firefox 17, 24, and 27, and for Google Chrome 30, 31, and 32.		x
For every presented vocable the user should be able to choose “Again”, “Good”, or “Simple”.	x	
The system must handle concurrent logins from 4,000 user.		x
If a user logins, the complete website must be presented within 4 seconds.		x
The command “calculate” will change the state of the map from green to red.	x	
The upload of a file must be finished after 20 seconds.		x

2.8 Describe the difference between “Component Integration Testing” and “System Integration Testing”. [/ 1]

- Component integration testing
  tests the interactions between software components and is done after component testing.
- System integration testing
  tests the interactions between different systems or between hardware and software and may be done after system testing.

3 Static Techniques [/ 9]

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

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

3.2 Which of the following is the main purpose in the review planning phase? [/ 1]

a) Planning who should be manager and scribe including time schedule ☐
b) Allocate the individual roles ☒
c) Log defects and motivate the reviewers to document more findings ☐
d) Gather metrics ☐

3.3 What is the main difference between Static testing techniques and Dynamic testing techniques? [/ 1]

- Static testing techniques rely on the manual examination (reviews) and automated analysis (static analysis) of the code or other project documentation without the execution of the code. Static testing is typically an early test activity.
- Dynamic testing techniques requires the execution of software.

3.4 Advantages of reviews [/ 1]

List 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 User Stories for requirements [/ 3]

Following requirement could be found in a specification:

> In an online pizza shop are two different kinds of customers defined. A normal guest generates a business volume up to 500 Thai Baht. A VIP guest generates a business volume of 500 Thai Baht or more. These VIP guests get discount for future orders about 5 %.
> Every interested person may order a menu-newsletter.
> An administrator should have the possibility to see how many people visited the online pizza shop and what business volume every guest generated.

Write 3 User Stories out of the requirement specification. Use following template:
As a <type of user>, I want <some goal> so that <some reason>.

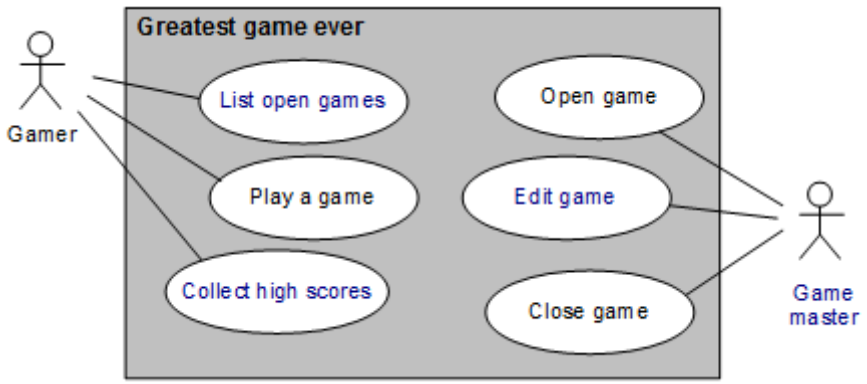
- As an interested person I want to order a menu-newsletter, so that I know what could be ordered.
- As a VIP guest I want to get 5 % discount, so that I like to order in future.
- As an administrator I want to see how many people visited the online pizza shop, so that I know how successful the online pizza shop is
- As an administrator I want to see what business volume a guest generated, so that I know my guest better.

3.6 Use Case Diagram [/ 2]

You got following requirements description:

> The application “Greatest Game Ever” offers for anyone to open, to edit and to close a game as a “game master”. Gamer could list open games, play a game and collect their high scores.

Enter the missing information in the Use Case Diagram below.



4 Test Design Techniques [/ 21]

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

Test design method	White box	Black box
Error guessing		x
Code coverage	x	
Design inspection	x	
State transition coverage		x

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

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

4.3 When should expected results for test cases ideally be defined? [/ 1]

a) when a defect was detected ☐
b) before tests get executed ☒
c) during test execution ☐
d) during test report generation ☐

4.4 Decision Tables [/ 4]

Following specification: Renting a car requires that a customer shows his driver license. It is possible to rent a Porsche as well. Preconditions for this are that the customer has a credit card and is 23 years or older. Paying with credit card, a customer gets 5 % discount.

a. Create a decision table based on the given specification [/ 3]

Car Rental Rules		1	2	3	4	5	6	7	8
Conditions	Customer has driver license	yes	yes	yes	yes	no	no	no	no
	Customer is 23 years or older	yes	yes	no	no	yes	yes	no	no
	Customer has credit card	yes	no	yes	no	yes	no	yes	no
Actions	Porsche car offer	x							
	Normal car offer	x	x	x	x				
	5 % discount	x		x					
	No car offer					x	x	x	x

b. How many test cases are required for full test coverage? [/ 1]

4

4.5 Equivalence classes [/ 1]

A software accepts days as numbers (1 = Monday, 2 = Tuesday, …, 7 = Sunday). Which values would cover all equivalence classes?

a) 0, 1, 2 ☐
b) 1, 2, 3, 4, 5, 6, 7 ☐
c) 0, 6, 14 ☒
d) 3, 6, 7, 14 ☐

4.6 Password example [/ 4]

Consider a software accepting for login only secure passwords.
The specification states that the password should have at least 8 characters and maximum 12 characters.
The password must content at least one upper-case letter (like “A, B, C”), one lower-case letter (like “a, b, c”), a number and a special character (like “!§$”). The password must not content a space character.
Additionally, if a new password is entered, it must be significant different from an older version, means it has to differ at least at three positions.

a. Add for each derived Equivalence Class below, if it is “valid” or “invalid”. [/ 2]

1. Password entered (valid)
2. No password entered (invalid)
3. Password is less than 8 characters in length (invalid)
4. Password is 8 to 12 characters in length (valid)
5. Password is greater than 12 characters in length (invalid)
6. Password contents upper-case letters (valid)
7. Password does not content upper-case letters (invalid)
8. Password contents lower-case letters (valid)
9. Password does not content lower-case letters (invalid)
10. Password contents a number (valid)
11. Password does not content a number (invalid)
12. Password contents a special character (valid)
13. Password does not content a special character (invalid)
14. The entry contains no blanks (valid)
15. The entry contains blanks (invalid)
16. Entering a new password, it differs at less than 3 positions (invalid)
17. Entering a new password, it differs at 3 or more positions (valid)

b. Write down for following test cases the expected outcome (T for TRUE or F for FALSE), and which equivalence classes are covered.
In case an old and new password is given, evaluate only the "*new password*". [/ 2]

Example:

Test case #	Test data	Expected outcome	Equivalence classes covered
1	*No password entered*	F	2

Test case #	Test data	Expected Outcome	Equivalence classes covered
2	`abcdefg`	F	1, 3, 7, 8, 11, 13, 14
3	`Abcdefgh!jk1m`	F	1, 5, 6, 8, 10, 12, 14
4	*Old password*: `Abcdefgh!jk1` *New password*: `abcdefghijk2`	F	1, 4, 7, 8, 10, 13, 14, 17
5	`D tY 10$$`	F	1, 4, 6, 8, 10, 12, 15

4.7 Equivalence Classes and Boundary values [/ 3]

A system was designed to calculate the fine in case of speeding. Speeding 1-15 mph over 65 mph results in a fine of $146. Additionally:

- Speeding 16-25 mph over 65 mph $266 fine.
- Speeding more than 25 mph over 65 mph $380 fine.

a. How many Equivalence Classes could be identified? [/ 1]

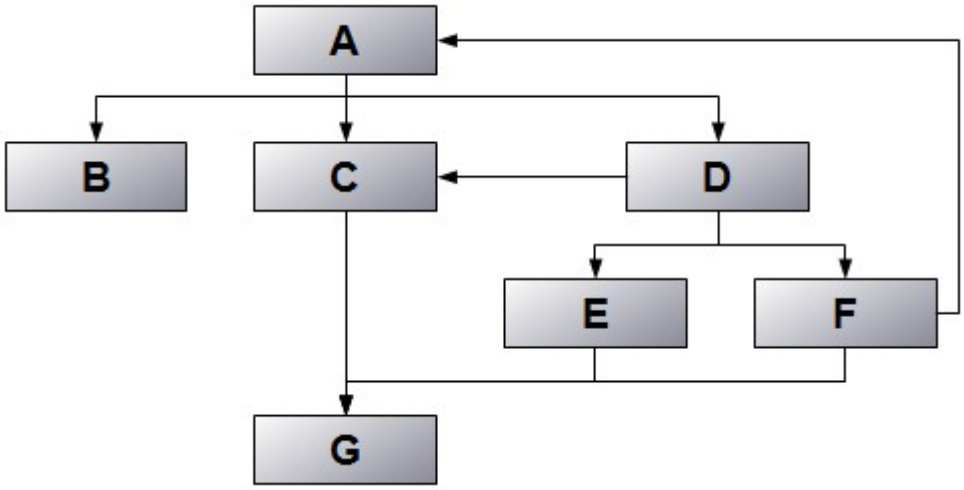
4

b. Write down all Boundary Values. [/ 2]

Speeding over 65 mph: 0 mph, 1 mph, 15 mph, 16 mph, 25 mph, 26 mph

4.8 Coverage and Code Complexity [/ 5]

Following graph is representing a program:



One test has been executed for the control flow graph shown above, it covers path: **A, B**.

a. How many tests are necessary to achieve 100 % Statement Coverage? [/ 1]

3

b. Write down, which paths the required tests should cover to achieve 100 % Statement Coverage. [/ 1]

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

c. How many tests are necessary to achieve 100 % Decision Coverage? [/ 1]

5

d. Write down, which paths the required tests should cover to achieve 100 % Decision Coverage. [/ 1]

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

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

M = L – N + 2 P = 10 – 7 + 2 x 1 = **5**