

# CS 5/7314: Software Testing & Quality Assurance

## Exam #1

Feb. 19, 2024

Student name:

Bingying Liang

Exam grade:

38/50

### General Instructions

- This is a closed-book exam, but one summary sheet (8x11, letter sized, single or double sided) is allowed.
- Read exam questions carefully before answering them.
- There are three questions, totaling 50 points. The point distribution is shown by each question.
- Exam duration for on-campus students (& distance students taking the exam on campus): 75 minutes.
- Distance students:
  1. Exam start time: You may start anytime after 2pm, 2/19/2024.
  2. Exam duration: 90 minutes total, including 75 minutes for taking the exam and an extra 15 minutes to handle the scanning/uploading/submission/etc.
  3. Deadline for exam submission: 90 minutes after you start your exam & before 3:30pm, 2/21/2024.
  4. You need around 3 sheets of blank paper to write your answers (no pencils, please, to ensure readability of the scanned exam).
  5. Be sure to clearly identify question numbers.
  6. When you finish answering the questions, you need to scan your answers and upload it to Canvas for submission.
  7. When upload your completed exam for submission, please try to use a single file, if possible.
- Good luck!

## I. Choose one best answer for each question ..... (3 points each, 15 points total.)

8/15

1. Consider the different quality attributes/characteristics we discussed in class:

- Reliability is always the most important.
- Safety is always the most important.
- Usability is always the most important.
- Security is always the most important.
- Maintainability is always the most important.
- All of the above.
- None of the above.

该经管以

(8)

0/3

2. This is NOT part of quality planning in the SQE process:

- setting the quality goals.
- determining the product design. *is not quality*
- determining the defect and other measurement/data to collect
- determining the analyses to perform.
- determining the set of QA activities/techniques.
- all of the above.
- none of the above.

该经管以

(b)

0/3

3. Match each of the following terms about testing and its description/definition:

- |  |  |
|--|--|
| <input checked="" type="checkbox"/> c i. test run              | a. testing the external functions          |
| <input checked="" type="checkbox"/> e f ii. test sensitization | b. a way to check/decide the test result   |
| <input checked="" type="checkbox"/> b iii. test oracle         | c. an execution instance                   |
| <input checked="" type="checkbox"/> a iv. BBT                  | d. testing the implementation details      |
| <input checked="" type="checkbox"/> d v. WBT                   | e. determining the input values for a test |

2/3

4. Consider the defect related terms: error, fault, and failure in testing:

- We observe errors and remove related faults.
- We observe errors and remove related failures.
- We observe faults and remove related errors.
- We observe faults and remove related failures.
- We observe failures and remove related errors.
- We observe failures and remove related faults.
- We directly detect and remove errors.
- We directly detect and remove faults.
- We directly detect and remove failures.

✓

3/3

5. Consider testing techniques PT vs Musa-OP:

- They are the same.
- They have nothing in common.
- PT model can be derived from Musa-OP by using uniform distribution of probabilities.
- Musa-OP can be derived from PT by adding usage probabilities to partitions.
- PT is based on partitions while Musa-OP is based on FSM.
- Musa-OP is based on partitions while PT is based on FSM.

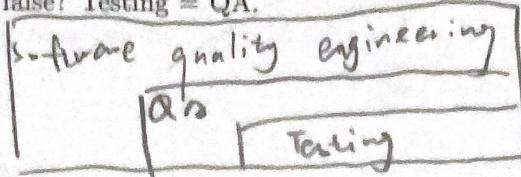
✓

3/3

## II. Testing, QA, and SQE (Briefly justify yourself.) ..... (5 points each, 15 points total)

1. True or false? Testing = QA.

False.



Ans: It is a subset of SQE focused on preventing defects and ensuring quality in the software development process. It involves activities and processes that assure the quality of software by working on methodologies, tools, and standards to manage, control, and improve the development process.

Testing: is a subset of QA. Testing primarily focus on the software's operation and user interaction, capable of uncovering functional and non-functional problems.

2. If you re-implement a software but keep the same functionality. Given a functional checklist and a module checklist for the old program, which one can be directly used to test the new software?

Example:

d. from

- ① abnormal termination
- ② installation
- ③ language
- ④ logging and recovery
- ⑤ different design
- ⑥ backup and restore
- ⑦ communication
- ⑧ co-existence
- ⑨ migration
- ⑩ gateway
- ⑪ stress
- ⑫ file I/O
- ⑬ index management

4/5-

Justify? ↗ ↘

if keep the same functionality  
the functional checklist can be  
directly used to test.

3. True or false? We can automate all the testing-related activities.

False. We just only can set realistic expectations and then test automation.  
specific areas for automation, esp. in execution, measurement,  
and analysis.

III. Simple Coverage and Usage Testing ..... 16/20 (10 points each, 20 points total)

1. For a software with two input variables,  $w$  for weight (light, medium, heavy) and  $h$  for height (tall, short). Construct your precisely defined PT model and test cases to test this software.

Sol.

partition 1: tall  $h=1$ , light  $w=0$

partition 2: tall  $h=1$ , medium  $w=1$

partition 3: tall  $h=1$ , heavy  $w=2$

partition 4: short  $h=0$ , light  $w=0$

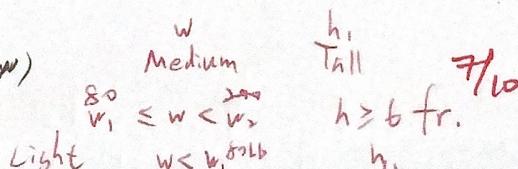
partition 5: short  $h=0$ , medium  $w=1$

partition 6: short  $h=0$ , heavy  $w=2$

Test case	Description	input	expected output
TC1	no input received	(empty)	Error message
TC2	tall, light	$h=1, w=0$	partition 1
TC3	tall, medium	$h=1, w=1$	partition 2
TC4	tall, heavy	$h=1, w=2$	partition 3
TC5	short, light	$h=0, w=0$	partition 4
TC6	short, medium	$h=0, w=1$	partition 5
TC7	short, heavy	$h=0, w=2$	partition 6

*Partitions need precise definitions*

I suppose  $light = 0$ ,  $medium = 1$ ,  $heavy = 2$  ( $w$ )  
 $tall = 1$ ,  $short = 0$  ( $h$ )



2. Now, assume that you have access to actual usage data (you may make up such data for the exam) for this software, construct a Musa-OP using either Musa-1 or Musa-2 OP construction procedures.

Sol: Musa OP - Musa-1 ✓  
① Customers of the software  
Business, Education, Government

Customer Type	Weight
Business	0.45
Education	0.10
Government	0.40
Other	0.05

## ③ System mode

Operational, Maintenance, Admin

System mode	Weight
Operational	0.90
Maintenance	0.08
Admin	0.02

## ④ Function: Code Generation

Identifying user,  
Accepting the code?  
Identifying Application  
Verify code correctness?  
Partition check?

② Users of the software  
Human Users, Programmers, Third Party

User type	User profile by Customer Type				Overall User profile
	Business	Education	Government	Other	
Human Users	0.80	0.80	0.80	0.10	0.7675
Programmers	0.10	0.20	0.20	0.10	0.1725
Third Party	0.05	0.00	0.00	0.75	0.06

9/10

⑤ Function

Function	Occurrence Rate	Usage
Code Generation	every input 30s	0.30
Identifying the code	every day	0.10
Accepting the code	every day	0.10
Identifying application	every day	0.10
Verify code correctness	every day	0.10
Partition check	every input 30s	0.30