جامعة الأمير سلطان
PRINCE SULTAN UNIVERSITY

College of Computer & Information Sciences Spring 2022 - Midterm Exam

SE401: Software Quality Assurance and Testing Time: 55 Minutes

Student Name: _____ ID: _____

Question	Weight	Your Score
1	6	
2	2	
3	4	
4	4	
5	4	
Total	20	

Question 1: Short Questions [6 points]

1) Why is it typically not possible to do exhaustive testing, meaning to test all inputs and preconditions?

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

2) Explain why the distinction between fault and failure is important when designing test cases.

   Failure, the user notices the system/program behaving incorrectly.

   Fault, this is a problem in the code, which is causing a failure in this case.

   A Software fault is a static defect in the software, whereas a software failure is an external, incorrect behavior with respect to the requirements r another description of the extended behavior if software. Faults are the basis for failures.

   In a medical example, faults are root cause of symptoms, whereas the failure are the symptoms itself.

   Distinguishing between them is very essential in designing the test cases based on your goal.

3) Explain the differences between verification and validation, and explain why validation is a particularly difficult process.

   Verification is involved in the conformance to the specification. The specified functional and non-functional requirements must be checked to meet the system needs. Validation is checking that the system meets the requirements of the customer. Validation is a difficult process because there are different stakeholders who have different requirements for the system. So, the system being developed needs gradual changes to meet the users' requirements. The identified needs when during the system specification stage may be different when the system undergoes testing.

Question 2: Multiple Choice Questions [2 points]

1) A system that fails to meet its user's needs may still be:
   a. Correct with respect to its specification.
   b. Safe to operate.
   c. Robust in the presence of exceptional conditions.
   d. Considered to have passed verification.

a,b,c,d

2) You are reading a user story in the product backlog to prepare for a meeting with the product owner and a developer, noting potential defects as you go. Which of the following statements is true about this activity?
   a. It is not a static test, because static testing involves execution of the test object
   b. It is not a static test, because static testing is always performed using a tool
   c. It is a static test, because any defects you find could be found cheaper during dynamic testing
   d. It is a static test, because static testing does not involve execution of the test object

d

Question 3 [4 points]. Answer the following questions:

1) Late in a project, after you have been testing for a long time, you find a serious bug that will require expensive revisions to an internal data structure. The project manager is furious because this type of change would have been much cheaper if the bug had been found much earlier. You both check, and determine that the bug was in the code before you ever started testing--you've missed it all these months. How would you decide whether missing this bug up to now was the result of inadequate testing, an erroneous testing strategy, or sensible conformance to a reasonable test strategy?

   This is a result of erroneous testing strategy. How their strategy did assume that the code is working without testing it. Assuming that the existing code is functional without issues is part of this erroneous testing strategy. Dependencies might cause the whole project to fail.

   It has nothing to do with inadequate testing since it is not part of the newly developed features/code.

2) A developer claims that because only a small proportion of software errors turn into software failures, it is unnecessary to make substantial investments in the prevention and elimination of software errors.

   a. Do you agree with this view?

   b. Discuss the outcome of accepting these views.

   Considering the damage caused by software errors, one must invest as much effort as possible in eliminating software errors to reduce these damages to a minimum. Some possible outcomes of the “no investments in errors prevention and detection” policy:

   - Lack of control over the extent of possible failures especially of severe failures.
   - A software failure may threaten a patient’s life or a soldier in combat.
   - Software failures are not limited to user’s waste of time until the failure is repaired, but cause damage to other systems.
   - Software failures occur at unexpected times and can cause maximum damage.

Question 4: Short Answers [4 points]

a. You have been asked to develop a test execution schedule for the second release of a customer data management application. The first release only allowed basic customer records to be added. This second one contains 5 new features and 2 bug fixes, together with the new features' priorities (1 is highest) and the defect severity levels, in table below.

Priority	Feature/Fix	Test Order
1	Delete inactive customer record	
2	De-activate customer	
3	Edit extra data items	
4	Edit Basic data	
5	Link customer records (for example in the same company)	
5	Provide extra data items when record added	

What would be the best test execution sequence for this release?

Priority	Feature/Fix	Test Order
2	De-activate customer	1
1	Delete inactive customer record	2
5	Provide extra data items when record added	3
3	Edit extra data items	4
4	Edit Basic data	5
5	Link customer records (for example in the same company)	6

The highest priority is to Delete an inactive customer, but first we must have an inactive customer, so we run De-activate customer first. This is a functional dependency.

The next highest priority is to Edit extra data items, but again, we need to have some extra data items in order to do this, another functional dependency.

The final two are independent so the priority 4 test is run before the priority 5 test.

b. Ahmad, a junior software tester, has just joined a very large online payment company in the kingdom. As a first task, Ahmad analyzed their past two years of bug reports. Ahmad observes that more than 50% of bugs have been happening in the International payments module.

Ahmad then promises his manager that he will design test cases that will completely cover the International payments module, and thus, find all the bugs.

Which of the following testing principles might explain why this is not possible?

1. Pesticide paradox.
2. Exhaustive testing.
3. Test early.
4. Defect clustering.

Exhaustive testing is impossible.

Question 5: Open-ended questions [4 points]

Have a careful look at the following (faulty) program:

```
public int findLast (int[] x, int y)
{
//Effects: If x==null throw NullPointerException
//   else return the index of the last element
//   in x that equals y.
//   If no such element exists, return -1
   for (int i=x.length-1; i > 0; i--)
   {
      if (x[i] == y)
      {
         return i;
      }
   }
   return -1;
}
```

a) Where is the fault and how should it be corrected?

i > 0 should be i >= 0
The current code will skip x[0]

b) Assume, you apply the following test cases TC1 and TC2:
(TC1) input: x = [2, 3, 5]; y = 2 – expected output: 0
(TC2) input: x = [2, 3, 5]; y = 3 – expected output: 1
Say for each test case whether it triggers a failure. Justify your answers.

TC1 ➔ Failure it will not check for x[0] so -1 will be returned
TC2 ➔ Pass it will give the desired result