



United International University

Department of Computer Science and Engineering

CSE 4495: Software Testing and Quality Assurance

Mid-term Examination : Fall 2022

Total Marks: 30 Time: 1 hour 45 minutes

Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.

Answer all the questions. Numbers to the right of the questions denote their marks.

1. (a) You want to build a website named *CodeWars* for UIU CSE students where they can host their own private programming contests among UIU students. In addition students can practice from a large collection of problems, add other users as friends, throw out challenges to each other etc. Now, with this vision in mind lay out a development plan and mention the verification and validation steps of your methodology. Who will participate in these V&V steps? What is the required level of V&V for your software? [2]
- (b) What is static verification and dynamic verification ? What are the advantages of one over the other? [1]
- (c) Describe any three of the following '*Quality Attributes*'. Use examples to illustrate possible specifications or requirements for these attributes. [3]
 - i. Usability
 - ii. Performance
 - iii. Scalability
 - iv. Security
2. (a) You manage an online resource sharing platform where people can share e-books and others study materials. A typical upload/download takes five minutes on average, and an interrupted upload/download must be restarted from the beginning. The number of customers engaged in a activity at any given time ranges from about 960-1200 during peak hours. On average, your system goes down (dropping all connections) about three times per week, for an average of six minutes each time. You are given the following two options to improve the availability of your system(so that less customers face interruption when they are using your services). [3]
 - i. Reducing total downtime to half (three minutes to restart on average after each crash).
 - ii. Increasing the current *MTBF* by 50% (i.e two system outage per week, but total downtime remains unchanged).Analyze both possibilities to prove statistically which method alleviates more users and decide which option better achieves your goal.
- (b) Imagine you are quality assurance lead of an online shopping service. While testing your company's upcoming mobile application you were provided the following requirements for product release- availability of at least 99.9% and *POFOD* less than 0.016 [3]

After the testing is done you receive the following report from the testing team - "During 7 days of testing the system processed 3750 requests.

 - i. 98.5% of all requests were successful .
 - ii. 57.1% of failed requests ended up causing a system crash.
 - iii. After each crash servers were restarted. Each restart took 12.5 minutes on average."Now depending on this report measure the availability and *POFOD* of your system. Also decide whether your product meets the requirements given for release.
3. (a) What do the terms '*Faults*' and '*Failures*' represent? Describe are the differences between these terms with an example. [1]
- (b) Consider the scenario described in Q.1(a). In this website you want to test the search and sort functions. The functionality of this feature is, it takes an input keyword from the user and displays a list of all problems which contain the keyword in their title. It also displays the problems in ascending order of difficulty. To test this feature which types of oracle(s) should a tester use? describe the design and purpose of these oracle(s) [3]
- (c) What is Unit testing? Describe types of Acceptance tests and their purposes. What is the reasoning behind executing Unit tests in far greater number than System level tests or UI tests? [2]

4. (a) The product comparison tool is a high-level function exposed by the API of an online shopping platform that is useful for many purposes. It is mainly intended to check the relevance of a certain product with another product in the database. For example, a product completely similar to another one. It might be partially similar or not at all similar as well. This service will determine such similarity or relevance between two products. [3]

ComparatorTool(Product productA, Product productB) returns comparisonCode;

A **Product** is a data structure consisting of:

- A unique identifying product code (String, four characters followed by three numbers).
- The product vendor (String, four character short code).
- The category that the product belongs to (String, e.g. Category - Electronics).
- The sub-category in the category (String, e.g. Category - Electronics , Sub-Category - Smartphones).

comparisonCode is an integer with value:

- 1 for *Perfect match*. That means both the products are made by the same vendor and belong to same category and sub-category.
- 2 for *Similarity match*. This means the products are from different vendors but belong to same category and sub-category.
- 3 for *Partial Similarity match*. This means the products are from different vendors and also belong to different sub-categories, but their category is still the same.
- 4 for *No match*.

Now, you have to design system level test cases using the category-partition method for the ComparatorTool function. To do this you have to-

- You are provided the following choices (aspects that you control and that can vary the outcome) for the two input products and the database.
 - Parameter: **productA**; choices- Product Code, Vendor, Category, Sub-Category
 - Parameter: **productB**; choices- Product Code, Vendor, Category, Sub-Category
 - Parameter: **Database**; choices- Number of Products, number of Categories, number of Sub-categories.
 - For each choice, identify a set of representative values.
- (b) Apply at least 5(five) **ERROR**, 2(two) **SINGLE** and 1(one) **IF** constraints to your representative values derived in Q.4(a). Calculate the total number of test specifications generated before and after applying these constraints. [3]
5. (a) What is *k-way* Combinatorial Interaction testing? Define a *3-way* covering set. [1+1=2]
- (b) Here are some of the configurations that can be controlled in a certain web browser: [4]

Allow Cookies	Warn About Add-Ons	Warn About Attack Sites	Warn About Forgeries
Allow	Yes	Yes	Yes
Restrict	No	No	No
Block			

Figure 1: Configurations for Question 5(b)

What will be the total number of test specifications in this case? Using Combinatorial interaction technique create a covering set that covers all 2-way combinations of these browser configurations.