SENG 5811 Homework #1

Date: 1/28/2023
Due: 2/03/2023
Topics: Testing Fundamentals
Problems: 3
Points: 40

This assignment is to be completed individually.
Please submit a PDF of your solutions on Canvas by the due date.

Problem I. 12 Points

Ignoring testing principles may lead to undesirable consequences such as the ones listed below. For each of the following, identify a principle (from the seven in the textbook) that is most likely to have been overlooked. Briefly (one or two sentences) explain your reasoning.

a) A verification engineer who had previously worked on safety-critical systems testing adopted testing techniques, which had served well in that domain, to test web applications at a new job. But this testing effort turned out to be consistently late and over-budget.

b) A software development organization that has a comprehensive suite of automated tests, runs those tests routinely and confirm that all tests pass before releasing updated versions of the software. Their latest update, however, caused a major embarrassment when an obvious and easy to catch defect was not detected resulting in a huge number of field reports on that single issue.

c) A large software development team following waterfall development process, carry out all their test activities in the verification stage of the waterfall process. Their testing effort was found to be much costlier than a competitor's, though, their products were similar in functionality and quality.

d) After a systematic campaign to complete every identified test objective for a project with no open or pending defect reports, a verification team manager truly believed and claimed that "the software is defect free". This perception was not shared by others, but the manager persisted only to be disappointed when reality stuck in the form of a user-filed problem report which led to the identification of a previously undetected defect.

Problem II. 14 Points

Textbook [GBV] in Chapter 1, Section 4, describes various test activities in a test process and their associated work products. In the following table, the left column enumerates work-products, and the right column enumerates some activities. In general, work-products may be needed as input or produced as output by activities.

a) Identify pairs of the form:

(work-product-number, activity-letter) that indicates a valid input relationship, and **(activity-letter, work-product-number)** that indicates a valid output relationship.

For each pair identified, briefly explain why you consider it to be a valid relationship.

1. Defect Reports	a. Test Analysis
2. Test Results and Logs	b. Test Design
3. Requirements Specification	c. Test Implementation
4. Test Cases	d. Test Execution
5. Test Conditions	e. Test Monitoring and Control
6. Source Code	f. Test Planning

b) Which of the above work-products are **testwares** (as defined in the textbook [GBV])?

Problem III. 14 Points

A program that we are tasked to test is expected to read in 3 integer values where each value can range from 1 to 200, consider the values as the lengths of the sides of a triangle and produce an output message that indicates what kind of triangle it is:

- **Equilateral** (all 3 sides equal)
- **Isosceles** (exactly 2 of the 3 sides are equal)
- **Scalene** (all 3 sides are of a different length)

Write a list of specific inputs that you deem *sufficient* to "thoroughly" test this program. Next to each set of inputs write the expected output. Feel free to "invent" more output messages as you need them.

Identify for each test case, a brief purpose for why that test data is included (It should typically complete the sentence of the form: "this test is included to exercise the possibility that…."). E.g., such a "test case" might look like the following:

INPUT	EXPECTED OUTPUT	PURPOSE
(1, 1, 1)	Equilateral	Triangle kind is equilateral

NOTE: Do not worry about any objective definition of “thoroughness” yet. Simply a sincere attempt to be thorough (as you interpret the word) is what is expected. Attempt to discover cases and behaviors that may have to be dealt with by the program in special ways but not explicitly stated above.