Department of Computing

Curtin University

Software Engineering Testing (SET)

Week 2 Laboratory/Tutorial

The following exercises are intended to be done in a laboratory/tutorial session with a teaching assistant or instructor present. The exercises have been designed to reinforce concepts taught in SET.

1. Discuss (in small groups) the role of Software Testing in Software Development, Maintenance and Operations.

2. You manage an online service that sells downloadable video recordings of classic movies. A typical download takes one hour, and an interrupted download must be restarted from the beginning. The number of customers engaged in a download at any given time ranges from about 10 to about 150 during peak hours. On average, your system goes down (dropping all connections) about two times per week, for an average of three minutes each time. If you can double availability or double mean time between failures (MTBF), but not both, which will you choose? Why? Discuss.
   [Hint: Think about "How Can We Assess the Readiness of a Product?" - Analysis and testing activities during development are intended primarily to reveal faults so that they can be removed. Identifying and removing as many faults as possible is a useful objective during development, but finding all faults is nearly impossible and seldom a cost-effective objective for a nontrivial software product. Analysis and test cannot go on forever: Products must be delivered when they meet an adequate level of functionality and quality. We must have some way to specify the required level of dependability and to determine when that level has been attained.
   Different measures of *dependability* are appropriate in different contexts. *Availability* measures the quality of service in terms of running versus down time; *mean time between failures* (MTBF) measures the quality of the service in terms of time between failures, that is, length of time intervals during which the service is available. *Reliability* is sometimes used synonymously with availability or MTBF, but usually indicates the fraction of all attempted operations (program runs, or interactions, or sessions) that complete successfully].

3. Multiple returns is a language feature that lets a method return more than one value. It allows you to avoid having a method return a collection when all you really want is to return one of two values and get them into variables, they can also complicate your testing. Consider the following algorithm:

   - What is the total number of paths?
   - How long would it take to execute all the paths?

- Would executing all paths completely test the program?
- Are all paths reachable?

```
public static int primeFactor(int n) {
    for (int divisor = 2; divisor < n; divisor++) {
       if ((n % divisor) == 0) {
          return divisor;  // Divisible implies not prime!
       }
     }
     return 0;  // Must be prime if nothing was able to divide it.
 }
```