Unit 3 Notes

**Learning Objectives**

* Apply combinatorial test coverage to assess test quality
* Apply design of experiments to develop tests
* Understand mutation testing
* Understand fuzz testing
* Define metamorphic testing
* Apply defect-based testing techniques
* Describe the role of exploratory testing

Combinatorial Testing Techniques

Combinatorial Coverage as an Aspect of Test Quality

* See slides

Design of Experiments

* See slides

Design of Experiments: Problem Example \*\*

A picture containing text, whiteboard

Description automatically generated

Combinatorial Testing Techniques Problem Example

* Combinatorial coverage looks at parameter values being individually tested.
  + False. Combinatorial coverage looks at how combinations of parameter values are tested together.
* Graphical user interface, text, application, email

  Description automatically generated
* What is the goal of design of experiments?
  + Minimize the number of tests we need to run. We are testing pairs of values for each input to minimize the number of tests.
* True or False. Design of experiments pairwise combination involves systematically testing all combinations of inputs.
  + False. Only pairs of values for each input are tested together, not all combinations of values of inputs.
* Given 3 inputs: P1 with values V1 and V2; P2 with values V3, V4, and V5 and P3 with values V6 and V7, what are the correct tests for a pairwise combination design of experiments?
  + Table

    Description automatically generated

Using Combinatorial Testing to Reduce Software Rework: Review of Reading

Combinatorial Coverage as an Aspect of Test Quality: Review of Reading

Mutation Testing

Mutation Testing

* See slides

Mutation Testing: Knowledge check

* What is NOT an example of a mutation?
  + Graphical user interface, text, application, email

    Description automatically generated

Mutation Testing: Review of Reading

Fuzz Testing

Fuzz Testing

* See slides

Fuzz Testing Knowledge Check

* True or False? Fuzz testing consists of random, invalid or unexpected inputs that are created automatically.
  + Fuzz testing is an approach to testing where invalid, random or unexpected inputs are automatically generated.
* True or False? Fuzz testing looks only for undesirable behavior or crashes.
  + Fuzz testing is not looking at specific inputs or outputs, but is instead looking for an error or a wrong behavior.

Metamorphic Testing

Metamorphic Testing

* See slides

Metamorphic Testing Knowledge Check

* True or False? Metamorphic testing makes the assumption that if there is a program with input x that results in output y, and there is a change to input x, that same change is not reflected in output y.
  + False. Metamorphic testing makes the assumption that when changes are made to an input, it is possible to predict changes on the output.
* Without using a calculator, what would be the expected output of this example using metamorphic testing for the third test case?:
  + Graphical user interface, text, application, email

    Description automatically generated

Defect Based Testing

Defect Based Testing

* See slides

Defect Based Testing Knowledge Check

* True or False? Defect based testing can only be applied at the unit level.
  + False. Defect based testing can be applied at any level of testing.
* True or False? Defect based testing looks to create test cases that target specific defect categories.
  + True. Defect based testing can target any defect category from the Beizer Generic Defect Taxonomy Categories.

Exploratory Testing

Exploratory Testing

* See slides

Exploratory Testing Knowledge Check

* True or False? In exploratory testing, all test scripts are not developed in advance.
  + True. What is tested next is based on the results of the previous tests.
* True or False? Exploratory testing focuses on a tour that helps detect a specific error.
  + True. Exploratory testing can consist of requirements, features, continuous use, documentation, etc. tours that focus on different errors.