Specification Based Testing – Part 2

Mutation Testing

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

Objective

Understand
Metamorphic
Testing

Test Oracle Problem

- A set of tests has been developed by an organization
- The organization executes the tests against a program
- All of the tests pass
- What can we conclude?

Mutation Testing

- Introduce defects (mutants) into program undergoing test
- Check to see if test cases can detect the mutant
- Work began in early 70's but was not widely adopted due to cost
- Today's automated testing environments make mutation testing feasible

Creating Mutants

- **Mutants are typically created via syntactical modifications of source code**
- **Mutation generation tools exist for this**

- **Examples of mutations**
  - Modify Boolean expressions (< vs <=)
  - Delete Statement
  - Modify Variable
  - Modify arithmetic operation

Mutation Testing Assumptions

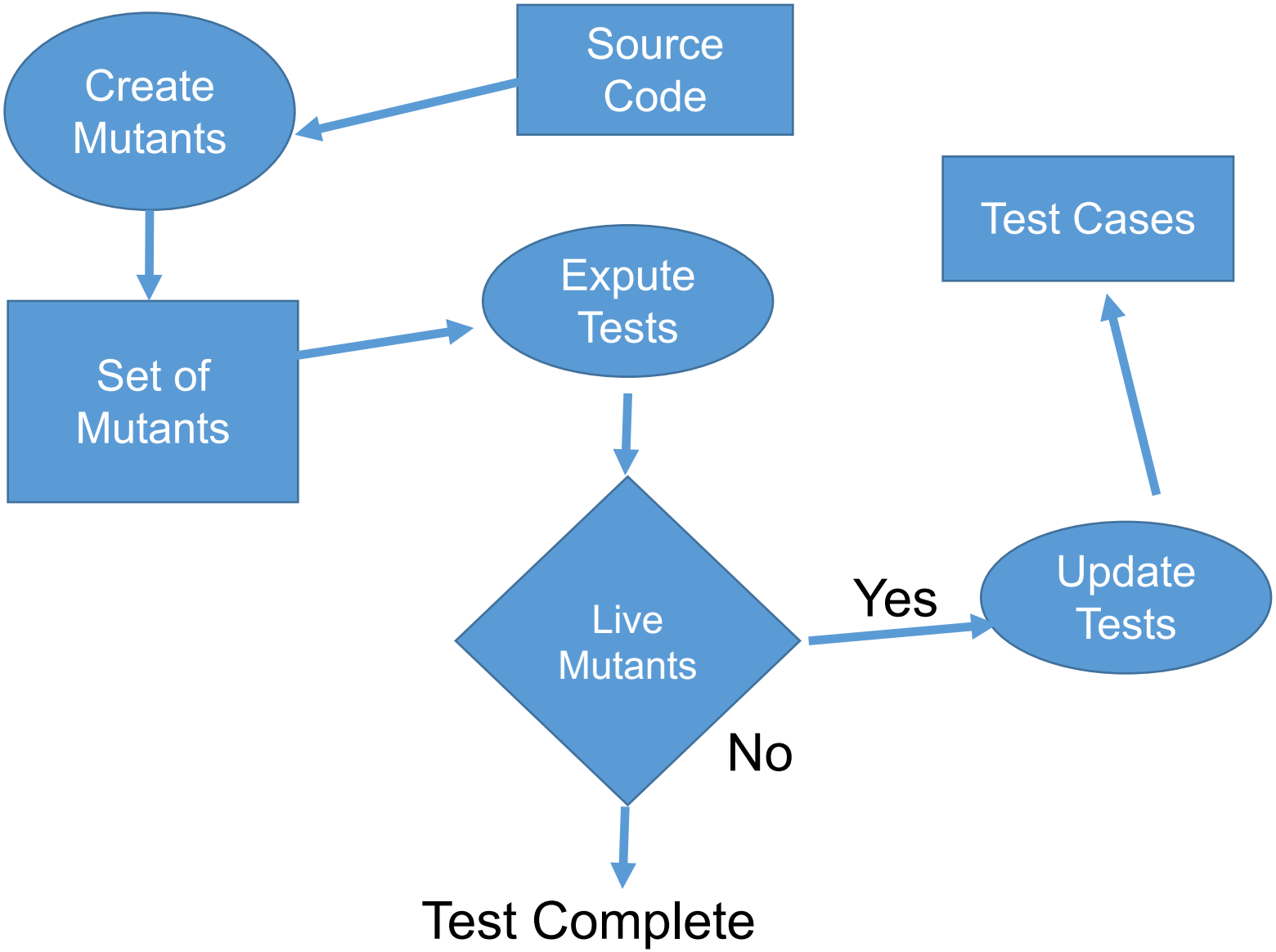
The Competent Programmer Hypothesis

- Programmers generally create code that is close to being correct reflecting only minor errors

The Coupling Effect

- Belief that test data that can detect small errors can also detect complex errors

The Mutation Testing Process



Summary