

Introduction to Software Testing
Chapter 9.2
Challenges in Testing Software – Software Testability

Paul Ammann & Jeff Offutt

www.introsoftwaretesting.com

Chapter 9 Outline

- 1. Testing for Emergent Properties:
Safety and Security**
- 2. Software Testability**
- 3. Test Criteria and the Future of
Software Testing**

Testability Overview

- Testability is distinct from software testing
- General definition :

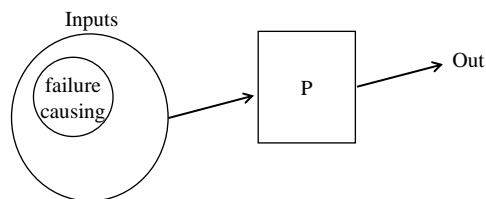
*An estimate or measurement of a conditional probability
– assuming that a software artifact contains a fault, how
likely is it that testing will reveal that fault ?*

- What do we do with testability estimates ?
 - Pay more attention to components with low testability – code reviews, formal analysis, stronger test criteria
 - Modify low testability components to increase their testability
 - We have more confidence in components with high testability
 - Risk analysis – low testability components represent risk that management needs to be aware of

Model of Testability

*If a program has a fault, how difficult will it be to
find a test that causes a failure ?*

Simple model

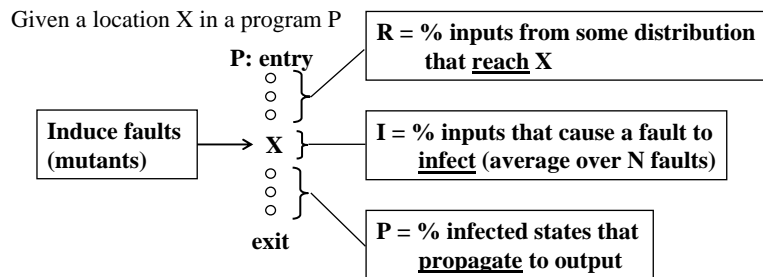


$$\text{Testability} = \frac{|\text{failure causing}|}{|\text{Input}|} \times 100 \%$$

Impractical to measure

Approximating Testability

Testability can be approximated with the RIP model and mutation



$$\text{Sensitivity (X)} = R * I * P$$

$$\text{Testability (P)} = F(\text{Sensitivity (X)}), \text{ for all X in P}$$

Issues in Approximating Testability

- Reasonable input distribution ?
- How to induce faults ?
 - What faults ?
- How to measure propagation ?
 - Expensive!
- Information hiding reduces propagation
- Assertion checking can be used to increase testability

Testability of OO Software

- **Information hiding decreases testability**
 - State information is saved in instance variables
 - No direct access to instance variables
- **Inheritance compounds the problem**
 - Instance variables are defined in ancestor classes – harder to get to
- **These are primarily issues of observability**
- **Increasing observability in OO software :**
 - Require additional **get ()** methods – must be done during development
 - Use Java reflection to access internal variables – hard to interpret the data values
- **This is an area of ongoing research**

Testability of Web Applications

- **Both controllability and observability are very low**
- **User interface and most software components distributed on different computers**
 - Server software may be distributed even further
- **Most communication is through message passing**
- **Much of the inter-component communication goes through the client**
 - Stateless HTTP messages
- **State is kept in an unusual combination of technologies**
 - Cookies and session objects

Testability in web applications is still very much an open research area

Testability Summary

- **Testability can give valuable information to testers, managers and developers**
- **Testability is often thought of as combining two characteristics of software**
 - **Controllability and observability**
- **Measuring testability is a very technical task**

How best to measure testability is still an open research problem