

INF 111 / CSE 121: Software Tools and Methods

Lecture Notes for Fall Quarter, 2007 Michele Rousseau Set 12

(Some slides adapted from Sommerville 2000 & Scott Miller)

Announcements

Quiz #2 – Still on Monday 10/29/07

- WILL INCLUDE:
 - All readings assigned since the last quiz not in the "WILL NOT INCLUDE LIST"
 - The lecture on 10/12 (except the MMM)
 - Lectures from 10/12 – 10/26 (today)
 - (Slide sets 7 through slide set 12)
- WILL NOT INCLUDE:
 - Ch 2 from "The Mythical Man-Month"
 - Van Vliet Ch. 4 will not be included on this quiz

Quiz #1 – Regrades need to be turned in by next Wednesday

- Come to my office hours tomorrow from 10:30-11:30a if you have questions on regrades
- Will regrades be accepted after Wednesday

Topic 12

2

Announcements (2)

Assignment #1 is due next week

- If you are having technical difficulties with Checkmate
- Make sure you notify me before the deadline
- Homework submissions made after the deadline will not be accepted
 - ...Without a good excuse
 - If you have a good excuse then it is essential you notify us prior to the deadline
 - ... if possible
- Please don't submit it the next day and say "I couldn't get checkmate to work here's my assignment" because
 - ... I can't fairly give you credit and I don't like having to not give you credit

Topic 12

3

Previously in INF 111...

- **More on Testing**
 - Test Adequacy
 - ▣ Coverage Based Testing

Topic 12 4

Today's Lecture

- **Equivalence Partitioning & Boundary Value Analysis**
- **Integration Testing**
 - Top-Down
 - Bottom Up

Topic 12 5

Test Adequacy

Tells you when to stop testing

- **Coverage-Based Testing**
 - Coverage metrics
 - ▣ when sufficient percentage of the program structure has been exercised
- **Fault-Based Testing**
 - Empirical assurance
 - ▣ when failures/test curve flatten out
 - Error seeding
 - ▣ percentage of seeded faults found is proportional to the percentage of real faults found
- **Error-Based Testing**
 - faults found in common are representative of total population of faults
 - Equivalence Partitioning

Topic 12 6

Equivalence Partitioning (EQP)

- Testing technique
 - Reduces the # of test cases
 - Make the # of test cases manageable
 - Systematic derivation of test cases
 - Reasonably tests the system

Basic Principle:
Some distinctions don't make a difference

Topic 12

7

EQP : How does it work

Divide inputs into **equivalent** partitions

- i.e. Find a small # set of **representative** input values
- For each Class program behaves in an "equivalent" way
- Smaller test set – but equally effective

Basic Method:
Notice when any element in the partition will produce the same results (ie find the same faults)

Topic 12

8

EQP: Reduces test cases

Input domain



Large set of test inputs.

Input domain partitioned into four sub-domains.



Four test inputs, one selected from each sub-domain.

Topic 12

9

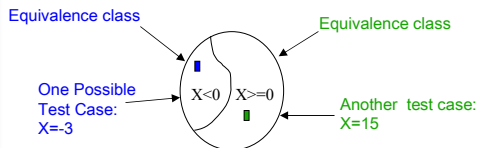
How to partition? Example 1

- Suppose that program P takes an input X, X being an integer.
- For $X < 0$ perform task (T1)
- For $X \geq 0$ perform task (T2)

Topic 12

10

Two sub-domains



- All test inputs in the $X < 0$ sub-domain are considered equivalent.
- The assumption is that if one test input in this sub-domain reveals an error in the program, so will the others.
- This is true of the test inputs in the $X \geq 0$ sub-domain also.

Topic 12

11

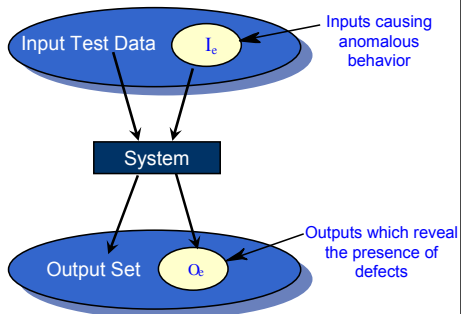
EQP: Basic Process

- First you must break the input into sub-domains (partitions)
 - Look at input and determine common properties
 - Values with in defined range
 - Values outside of the defined range
 - Extremes
 - Try to include input that will force incorrect output
 - How well does the code perform exception handling
- If the sub-domains are well done
 - should be able to create a few (or ideally) one test case that will represent the entire domain

Topic 12

12

Include inputs in and out of range



Topic 12

13

EQP: Example 2

- **Input should be a numerical month**

- Valid Inputs: 1-12

- **What are potential Classes?**

- Input within range:
 - 1-12
- Out of Range
 - High End: 20, 99, 3-digit, 4-digit
 - Low End: Negative Numbers
 - Alphanumeric
 - Special Characters / Punctuation

Topic 12

14

Boundary Value Analysis (BVA)

- **Select test cases based on the boundaries values**

- **Look for inputs**

- On the boundary
- On either side of the boundary

- **For numeric month example**

- Boundary Values
 - Low End: 0,1,2
 - High End: 11,12,13

- **Combining this technique with Equivalence Partitioning is much more effective**

Topic 12

15

EQP & BVA

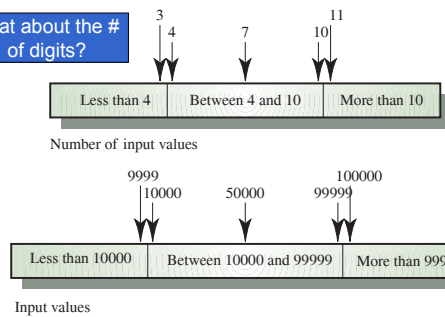
- **Input**
 - 5-digit integer between 10,000 and 99,999,
- **Partitions**
 - <10,000
 - 10,000-99,999
 - > 10, 000
- **Boundary Values**
 - 00000
 - 09,999 –10,000
 - 99,998 – 99,999 – 100,000
- **Outside**
 - Alphanumeric
 - Symbols

Topic 12

16

Equivalence partitions

What about the # of digits?



Topic 12

17

Pros & Cons

- **Pros**
 - Minimizes test cases & maintains some test adequacy
 - Forces tester to analyze the input and output domain
- **Cons**
 - Assumptions of equivalence can be tricky – and incorrect
 - Doing EQP is easy → Doing it well is difficult
 - Somewhat subjective – dependent upon the testers' intuition

Topic 12

Picking the correct subdomain can be tricky

18

Today's Lecture

- **Equivalence Partitioning & Boundary Value Analysis**
- **Integration Testing**
 - Top-Down
 - Bottom Up

Topic 12

19

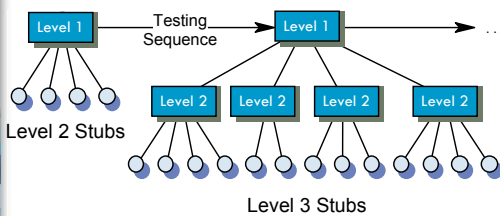
Integration Testing Approaches

- **Top Down & Bottom Up**
 - Top-down integration testing →
 - better at discovering errors in the system architecture
 - allows a limited demonstration at an early stage in the development
 - Bottom up →
 - Often easier to implement
- **Problems with both approaches. Extra code may be required to observe tests**

Topic 12

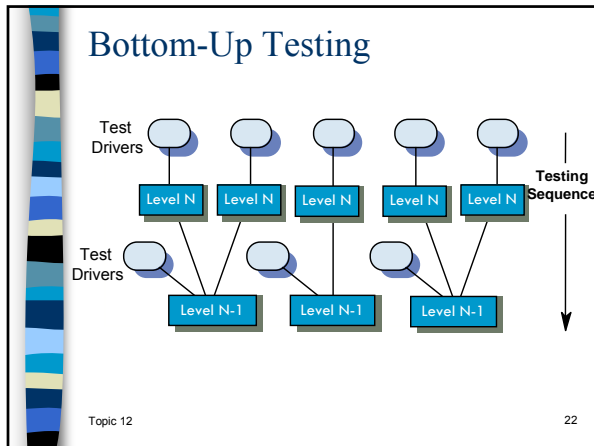
20

Top-Down Integration Testing



Topic 12

21



Which Approach to use?

- **Top-Down or Bottom Up?**
- **In practice, most integration involves a combination of these strategies**

Topic 12 23
