IS2545 Lecture 2

Testing Theory and Terminology

What is testing?

Fundamentally, comparing what an application is expected to do with what it actually does.

Example

If you write a program that calculates the square root of a number, what would you expect it to return when you provide the number 4 as input?

3? -1? null?

How much testing is enough?

We want to insure that our square root program behaves correctly, no matter the input.

Do we have to test every possible input?

-∞…-2, -1, 0, 1, 2…∞

Equivalence Class

A group of input values which provide the same, or similar type, of output.

Example

In our square root example, 1, 2, 3…∞ is one equivalence class.

Equivalence Class Partitioning

Separating a specific functionality into distinct equivalence classes based on input values.

Example

-∞…-3, -2, -1	imaginary numbers
0	0
1, 2, 3…∞	positive numbers

Another example

For a sporting goods store:

- If an item is discounted, add the word ‘Sale’ to the item’s title.
- If an item is discounted more than $10, display the sale price with the original price ~~striked through~~.
- If an item is discounted more than $20, do not display the price. The customer must add the item to their shopping cart to see the price.

Equivalence Classes

Item is not discounted.

Item is discounted by $10 or less.

Item is discounted by more than $10 but less than $20.

Item is discounted by more than $20.

Partitions

We have 3 partitions.

1. The discount partition.
   a. Item is discounted.
   b. Item is not discounted.
2. The strikethrough partition.
   a. Item is discounted by less than or equal to $10.
   b. Item is discounted by more than $10.
3. The shopping cart partition
   a. Item is discounted by less than or equal to $20.
   b. Item is discounted by more than $20.

Use equivalence classes to minimize testing efforts.

Instead of testing items that are discounted $0, $1, $2…∞, we can significantly decrease the number of tests we need to run by testing at least one value from each equivalence class.

1. $0 - Item is displayed with no changes.
2. $1 - The word ‘Sale’ is added to the item’s title.
3. $11 - The word ‘Sale’ is added to the item’s title and the original price is displayed and striked through.
4. $21 - The word ‘Sale’ is added to the item’s title and the price can only be displayed by adding it to your shopping cart.

Boundary Values

Selecting just one item from each equivalence class is often not enough to insure high quality.

Defects are more likely to occur at the boundaries of equivalence classes.

Boundary values of our discount example

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, ∞

Interior values of our discount example

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, ∞

Implicit Boundaries

Some boundaries are defined by the architecture or hardware of the system under test.

- MAXINT & MININT.
- Floating point precision.
- Memory size of the system.
- Null values.

Types of test cases

- Base case (happy path)
  - A test whose input is within the expected parameters of normal use.
  - Many of the values from the equivalence classes of the discount example are base cases.
- Edge case
  - A test whose input is not necessarily an expected value.
  - For example, requesting a discount of a negative value is an edge case.
- Corner case
  - A test whose input is “ludacris”.
  - For example, requesting a discount of an item that doesn’t exist.

Black-box testing

Testing the code as a user would, with no knowledge of the codebase.

Most manual tests are black-box tests. Such as executing a test via a web browser.

White-box testing

Testing the code directly and with full knowledge of the code under test.

Unit testing is an example of white-box testing.

Grey-box testing

Testing the code as a user would, but with knowledge of the codebase in order to understand where errors might be hiding.

A mixture of white-box testing and black-box testing.

Example:

- During a code review, you notice that a bubble sort is used. You then write a test that targets bubble sorts poor performance

Dynamic testing

Testing the system by executing it.

Static testing

Testing the system without executing any of its code.

- Code reviews
- Static code analysis