A Quick Introduction to Domain Testing

CAST 2019 Workshop
August 9, 2019

@ckenst @n00btester

Exercise 1 - Variable Tour

- Charter: Tour the Mortgage-Calc application looking for what you can change. Anything you can change is a variable. Find them all and make a list. Select one interesting variable and write down what values it can take.
- Timebox: 20 Minutes

Debrief - Put Example Variable Tour here

- This should be a table with of the input variables, relationship to output variable? Values?
- Classical table?
- All variables?

This Workshop

- We'll start with the basics
  - Then try to get more elaborate and difficult as time allows
- Team Up
  - Teams are highly encouraged
- Practice using our 2 target applications:
  - https://www.cken.st/mortgage-calc/
  - http://chrisreads.herokuapp.com
- Mostly taking a black-box approach but you of course could apply this in a more white-box way.

Testing with Variables - Revise

- Variables: a value or object that a program interacts with.
- Data Types: Programmers define the types of their variables
  - Integer
  - Boolean
  - Floating Point
  - Strings
  - Enumerated values
  - Byte
  - Char

Objectives - delete

- Practice applying Domain Testing
  - We develop skill over time, with practice and feedback.
  - We hope this workshop provides some practice and feedback
- Identify and characterize data types used by variables in the application
- Determine variables that are good candidates for domain testing
  - When looking at an application be able to exercise variables using data
- Generate best representative data and put that information into a domain testing table

Straw Poll - delete

1. How many people have heard of Domain Testing before?
2. What about Boundary Analysis or Equivalence Class Partitioning?
3. How many people here have applied either Domain Testing before (or Boundary Analysis or Equivalence Class Partitioning)?
4. How many people have heard of the term Quicktests or Quick attacks?
5. Has anyone taken the BBST Test Design course?

What are Quicktests or Quick Attacks? - delete

- Quicktests or Quick attacks are tests that don’t cost much to design, are based on some guesstimate for how the system could fail (risk-based) and don’t take much prior knowledge to apply.
  - They are great for starting your testing but can run out of steam quickly.
- https://searchsoftwarequality.techtarget.com/tip/Ten-quick-attacks-for-web-based-software

Test Design Overview

- Different tests reveal different kinds of information
- Techniques are a way to guide our thinking as we develop a group of tests
  - Think of functional testing, specification testing, domain testing, et. al as techniques
- One technique might be good at uncovering certain kinds of information but be bad at revealing other kinds of information
  - The best test strategies often contain various techniques, not just a single one

Test Design Overview

- Risk-Based Testing is about thinking of ways in which the application might fail and then testing to see if it does.
  - Risks can include or describe a way the program can fail (or has failed)
- Complete testing is impossible
  - Any sufficiently complex application will have too many possible values to test
  - Too many inputs for us to test with a reasonable amount of time

What is Domain Testing?

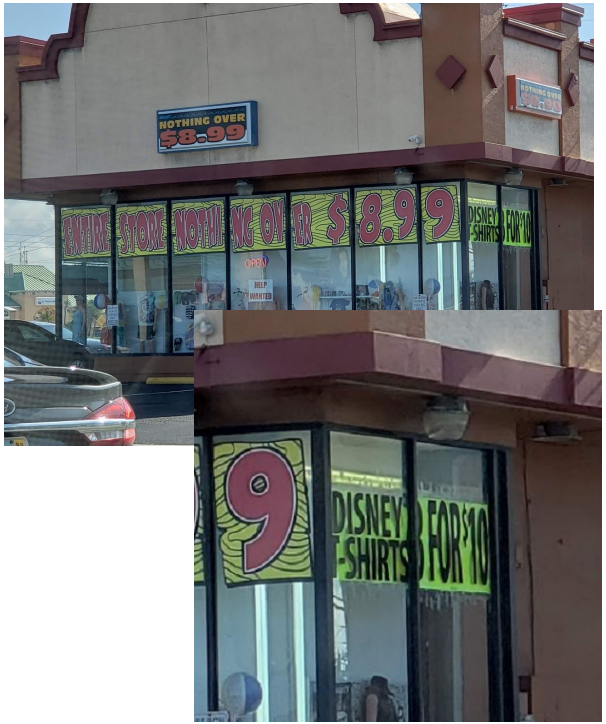
- In mathematics a Domain is a set of possible values for a function
  - Input domain
  - Output domain
- In software development, Domain Testing is an umbrella term for:
  - Boundary analysis
  - Equivalence class partitioning
  - Depending on the test design book you are reading you might this might also be called Domain Analysis or Input Domain Analysis.
- We combine them because these separate techniques often go hand in hand.

What is Domain Testing?

- It’s a risk-based sampling strategy for efficiently hunting for bugs in a programs handling of data
  - Anytime you have a very large set of data you can use and want to reduce it to a smaller sample size
- We don’t want to have to run tests for every value, in many ways we can’t
- Instead we can select a few values we think will uncover issues

What is boundary analysis? - Revise

- Boundaries are exactly what you think they are.
- Think of the Goldilocks rule:
  - Too big
  - Too small
  - Just right;
  - But also.. Too strange?
- Example:
  - Integer Value (Int32)
    - Too Big: 2147483648
    - Too Small: -2147483649
    - Just Right: 0 or 1 or 1000
    - Too Strange: A#$@! or (2-1) or 0.01 or +=1



Variables for MortgageCalc - Doesn’t look pretty!!!

Input Variables:
Home Value
Loan Amount
Down Payment
Interest
Term
Property Tax
HOA Amount
Extra Payment

Result Variables:
Monthly Taxes & Fees = (HomeValue * PropertyTax / 12)
Total Payment = Monthly Mortgage + Taxes & Fees)
Monthly Mortgage = (MonthlyRate * LoanAmount * (1 + MonthlyRate) ^ Number of MonthlyPayments) / ((1 + MonthlyRate) ^ NumberOfMonthlyPayments) - 1

Storage Variables:
NumberOfMonthlyPayments = Term * 12
MonthlyRate = Interest/12/100

https://en.wikipedia.org/wiki/Mortgage_calculator#Monthly_payment_formula

More on Variables - Revise

- We work with this idea of *notational variables*
  - We analyze the program as if these variables and our understanding of them is correct but it may not be
- Introduce the concept of input and result variables
  - Calculations should add up
  - Where is the data going?
  - Also something called a storage variable
- Can you get past the input filter?
- Do certain variables constrain other variables

Talk about Data Types - Revise

- Some data types are more applicable than others…
- Good for Domain Analysis
  - Integers
  - Floating Point
  - Strings
  - Byte
  - Char
- Not good
  - Binary
  - Boolean
  - Enumerated

What is an equivalence class? - Revise

- A set of values we think the program will handle the same way
  - If one catches a bug, they all will
  - If one doesn’t catch a bug, the others probably won’t either
- We partition a domain (input or output) based on how we think the application will handle it.
  - Valid Values
  - Invalid Values
  - Primary Domain
  - Secondary Domain

What is an equivalence class? - Revise

- Numbers are often used as a simple example
  - Use an example from one of the books
- However numbers are easy to find the equivalent of and they might also be the boundary values.
- When are equivalence classes hard? What data types?

What is a best representative? - Revise

- A best representative of an equivalence class is a boundary or another value in the class that seems more likely than other values to cause a program failure
- They all deal with data. In many instances, like with numbers (integers) your best representatives will probably be boundary values.

When do we apply Domain Testing? - Optional

- Combined with another technique
- You might use an ET Tour focused on understanding the variables of a system during which you might try to apply this.

Classical Table - Revise

- We need to bring this up at some point.
- How do we integrate it?

Exercise 2 - One Good Variable

- Charter: Tour the application with the goal of identifying one variable you think is a good candidate for domain testing. Using equivalence classes and boundaries identify the best representative data.
- Timebox: 30 minutes

Debrief - Revise

Risk-Based Testing

- Risk Equivalence Table

Scratching the Surface

- Each domain can be partitioned based on primary and second dimensions
- 18 steps for the Domain Testing Schema

References

- Domain Testing:
  - Kaner, Cem - BBST Domain Testing Workbook
  - Kaner, Cem - BBST Test Design Workbook
  - Copeland, Lee - A Practitioner’s Guide To Software Test Design
  - Beizer, Boris - Software Testing Techniques
  - Jorgensen, Paul - Software Testing: A Craftsman Approach, 3rd Edition
- Quicktests:
  - https://blog.gurock.com/beyond-quick-attacks/
  - https://searchsoftwarequality.techtarget.com/tip/Ten-quick-attacks-for-web-based-software
  - https://www.kenst.com/2018/05/what-are-quicktests-and-when-are-they-used/
  - Heuristic Test Strategy Model
  - Hendrickson’s Cheat Sheet

Contact Information

Everything past here is discarded

What is Domain Testing? - Revise

- What does this mean?
  - Domain Testing is an umbrella term for a number of techniques such as boundary analysis and equivalence class partitioning. Depending on the test design book you are reading you might this might also be called Domain Analysis or Input Domain Analysis.
- What else should we include?
  - Should we bring up that we're going to be reviewing a shorthand version of Domain Testing as there are 18 potential steps to Domain Testing - from "The Domain Testing Workbook"?
  - Characterize the Variable
  - Analyze the Variable and Create Tests
  - Generalize to Multidimensional Variables
  - Prepare for Additional Testing

Simplify

We need to keep this simple and introduce some of this other stuff after the exercise.

- Re-hash test desin
- Then introduce this concept of variables, input, storage and output
- Then introduce the concept of data types. Maybe?
- Then the Variable Tour exercise

Do we go on, or stop here?

- We should continue to build this out and get deeper.
- This will be time dependent

How Dwayne Pictures the Workshop Flow:

Introductions - 5 min
Overview and Objectives - 5 min
Intro into Domain Testing concepts - why it’s important - 10 min
Practice #1 - Chris Reads - 20 min
Debrief, introduce variables that have an impact on other variables and result variables - 10 min
Practice #2 - Mortgage Application - 20 min
Debrief, ask about key takeaways and any other questions - (15 - 20 min)

Workshop Goals & Notes (Do not publish)

- Need more examples
- Need more graphics
-

Exercise 1

- Charter: Tour the Mortgage-Calc or Book List applications, looking for data that might cause the application to fail. Make a list of the data you used along with what fields you used them on / tested.
- Timebox: 20 Minutes