



Reducing the Cost of User Acceptance Testing with Combinatorial Test Design

PNSQC - 2014

Steve Dyson
October 21, 2014

The Challenges

- User Acceptance Testing (UAT) is part of our typical testing strategy.
- UAT is often a manual process making it expensive, time consuming, and heavily reliant on shared business resources.
- Long UAT cycles have also led to delayed deployment schedules.
- We had no test automation in place for UAT
- Existing test suites may not be optimal

The Challenges

- User Acceptance Testing (UAT) is part of our typical testing strategy.
- UAT is often a manual process making it expensive, time consuming, and heavily reliant on shared business resources.
- Long UAT cycles have also led to delayed deployment schedules
- We had no test automation in place for UAT
- Existing test suites may not be optimal
- How could we gain efficiencies and reduce the amount of time needed for UAT?

The Challenge

In 2012, Cambia Health Solutions hired IBM to help identify solutions for finding process efficiencies in our projects in preparation for the federally mandated upgrade to ICD-10, which is the reclassification of all diagnosis codes used in medical billing.



COMBINATORIAL TEST DESIGN

Combinatorial Test Design

A National Institute of Standards & Technology (NIST) study concluded that the majority of software defects occurred because of the interaction between two variables



Combinatorial Test Design

Combinatorial Test Design (CTD), or Pairwise Testing, is a test modeling approach that takes the parameters of a software application and combines the interactions between those parameters into an optimized set of test cases based on a defined level of interaction.

Benefits of CTD

- Create a large test suite from scratch without manually writing each test case
- Identify testing gaps in existing test suites
- Eliminate duplicate test interaction in existing test suites
- Allows testers to execute test cases more frequently

Combinatorial Test Design

- Parameter – a measurable factor of the system under test

Ex. Browser

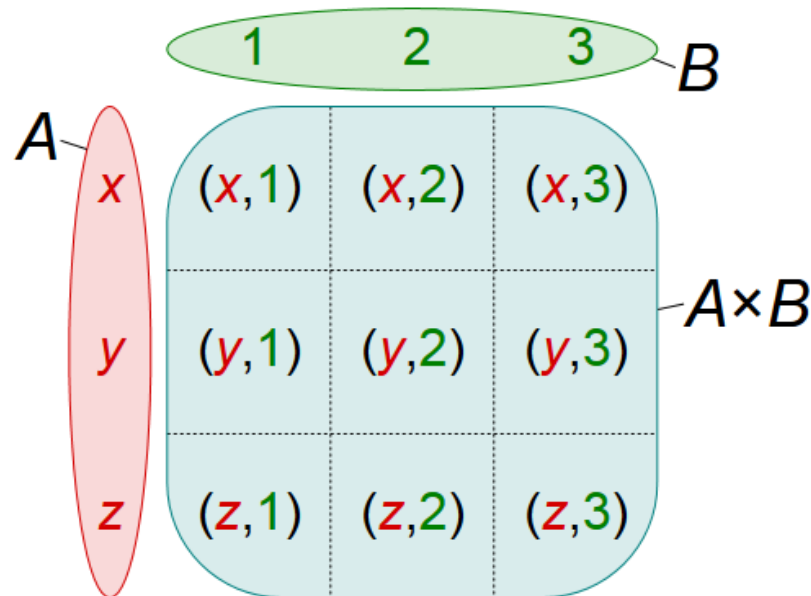
- Value – a characteristic of the parameter

Ex. – Chrome, IE, Firefox, etc.

How CTD Works

Cartesian Product

- A mathematical operation which returns a set from multiple sets.



How CTD Works

The CTD tool creates the Cartesian Product based on your parameters and values, and creates the lowest number of possible test cases by removing duplications and combining interactions

ATTRIBUTES



TEST CASES

CTD Example

Medicare Insurance Claims

CTD Example

Parameters

- Gender
- Primary or secondary insurance policy
- Claim Payment Status
- Correct Claims Submission
- Prior Documentation
- Claim Type
- Patient on Social Security
- Patient Disability Status
- Patient Work Status

CTD Example

Taking our test requirements and writing them in a parameter & value format simplifies the import/entry into the CTD tool

Gender	Coverage	Claim Payment Status	Claim form	Documentation	Claim type	SSA	Disabled	Working Status
Male	Primary	Fully paid	Properly formatted	Record of insurance	Professional	SSA	Disabled	Working
Female	Secondary	Partially paid	Improperly formatted	No record of insurance	Medical	No SSA	Not Disabled	Not Working
		None paid			Dental			

CTD Example

With this simple example, there are 1,152 potential interactions that need to be accounted for full test coverage

Gender	Coverage	Claim Payment Status	Claim form	Documentation	Claim type	SSA	Disabled	Working Status
Male	Primary	Fully paid	Properly formatted	Record of insurance	Professional	SSA	Disabled	Working
Female	Secondary	Partially paid	Improperly formatted	No record of insurance	Medical	No SSA	Not Disabled	Not Working
		None paid			Dental			

CTD Example

So much duplication!!!

Gender	Coverage	Claim Payment Status	Claim Form	Documentation	Claim Type	SSA	Disabled	Working
Male	Primary	Fully Paid	Properly Formatted	Record of Insurance	Professional	SSA	Disabled	Working
Male	Primary	Fully Paid	Properly Formatted	Record of Insurance	Professional	SSA	Disabled	Not Working
Male	Primary	Fully Paid	Properly Formatted	Record of Insurance	Professional	SSA	Not Disabled	Working
Male	Primary	Fully Paid	Properly Formatted	Record of Insurance	Professional	SSA	Not Disabled	Not Working
Male	Primary	Fully Paid	Properly Formatted	Record of Insurance	Professional	No SSA	Not Disabled	Not Working
Male	Primary	Fully Paid	Properly Formatted	Record of Insurance	Professional	No SSA	Not Disabled	Not Working

CTD Example Results

After running our simple example through a CTD tool, we are able to combine 1,152 interactions into 10 test cases

Test Number	Gender	Coverage	Claim Payment Status	Claim form	Documentation	Claim types	SSA	Disabled	Working
1	Male	Primary	Fully paid	Properly formatted	Record of insurance	Professional	SSA	Disabled	Working
2	Female	Secondary	Partially paid	Improperly formatted	No record of insurance	Professional	No SSA	Not Disabled	Not Working
3	Female	Primary	Fully paid	Properly formatted	No record of insurance	Medical	No SSA	Disabled	Not Working
4	Male	Secondary	None paid	Improperly formatted	Record of insurance	Professional	SSA	Not Disabled	Working
5	Female	Secondary	Fully paid	Properly formatted	Record of insurance	Dental	No SSA	Not Disabled	Working
6	Male	Primary	Partially paid	Improperly formatted	Record of insurance	Medical	SSA	Disabled	Working
7	Male	Primary	Partially paid	Properly formatted	No record of insurance	Dental	SSA	Disabled	Not Working
8	Male	Secondary	None paid	Properly formatted	No record of insurance	Medical	No SSA	Disabled	Working
9	Female	Primary	None paid	Improperly formatted	Record of insurance	Dental	SSA	Not Disabled	Not Working
10	<i>Male</i>	<i>Primary</i>	Fully paid	Improperly formatted	<i>No record of insurance</i>	Medical	<i>SSA</i>	Not Disabled	<i>Not Working</i>

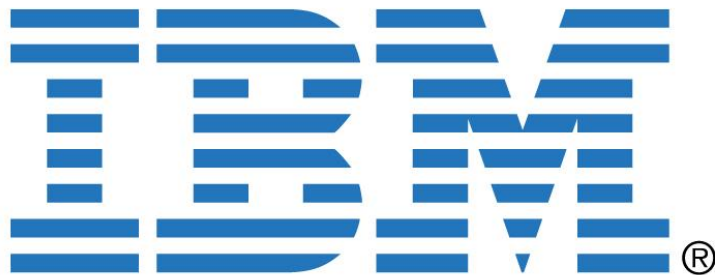
CTD TOOLS

CTD Tools



Testcover.com

Your test design requirements. Covered.™



CTD PROCESS

CTD Process Phases

1. Discovery & Test Planning
2. Entering Parameters
3. Adding Restrictions
4. Setting the Interaction Levels
5. Executing Test Case Creation
6. Test Case Coverage Analysis

1. Discovery Phase

We need to identify a few things...

- Is there variability in the system?
- Is the system structured in a parameterized fashion?
- Is the system complex enough that we need to actually optimize the test suites?



2. Entering Parameters

Edit Parameter

Parameter Name: ▼

Values (each value on a new line):

Medicare Claims Example [Bulk Edit](#)

Gender (2)	Male	Female	
Coverage (2)	Primary	Secondary	
Claim Payment Status (3)	Fully paid	Partially paid	None paid
Claim form (2)	Properly for...	Improperly f...	
Documentation (2)	Record of in...	No record of...	
Claim types (3)	Professional	Medical	Dental
SSA (2)	SSA	No SSA	
Disabled (2)	Disabled	Not Disabled	
Working (2)	Working	Not Working	

3. Adding Restrictions

Medicare Claims Example

Paired Values

never Disabled = Disabled
 never Working = Working



Value Expansions

Notes

Create New Note

simple test plan related to insurance

Medicare Claims Example Bulk Edit			
Gender (2)	Male	Female	
Coverage (2)	Primary	Secondary	
Claim Payment Status (3)	Fully paid	Partially paid	None paid
Claim form (2)	Properly for...	Improperly f...	
Documentation (2)	Record of in...	No record of...	
Claim types (3)	Professional	Medical	Dental
SSA (2)	SSA	No SSA	
Disabled (2)	Disabled	Not Disabled	
Working (2)	Working	Not Working	

4. Setting The Interaction Levels

Strength

2-way interactions ▼

[2-way interactions means...](#)

Hexawise tests cover all possible 2-way interactions in just: **10 tests** of a total possible **1,152 tests**

Strength

2-way interactions
3-way interactions
Mixed-strength interactions

[2-way interactions means...](#)

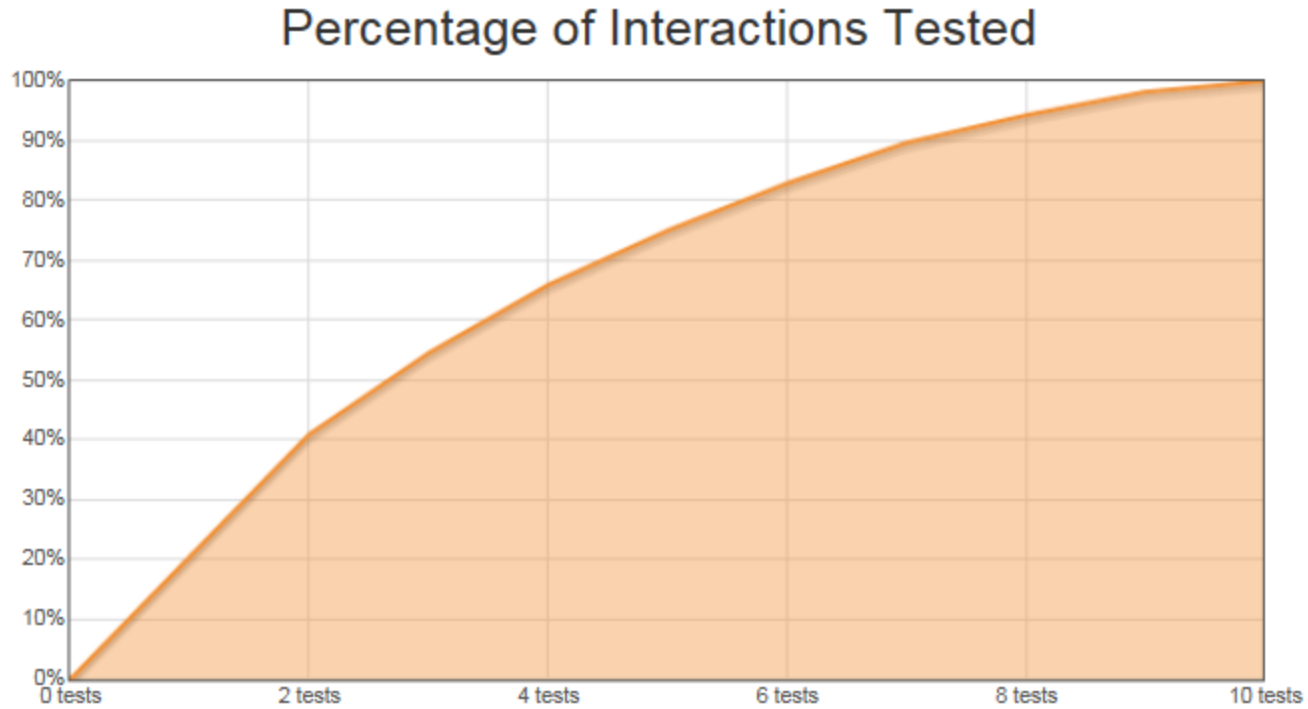
Hexawise tests cover all possible 2-way interactions in just: **10 tests** of a total possible **1,152 tests**

5. Executing Test Case Creation

Medicare Claims Example (2-way interactions) - values in purple *italics* can be replaced with any valid value

	Gender	Coverage	Claim Payment Status	Claim form	Documentation	Claim types	SSA	Disabled	Working
1	Male	Primary	Fully paid	Properly formatted	Record of insurance	Professional	SSA	Disabled	Not Working
2	Female	Secondary	Partially paid	Improperly formatted	No record of insurance	Professional	No SSA	Not Disabled	Working
3	Female	Secondary	Fully paid	Properly formatted	No record of insurance	Medical	No SSA	Disabled	Not Working
4	Male	Primary	None paid	Improperly formatted	Record of insurance	Professional	SSA	Not Disabled	Working
5	Female	Primary	Fully paid	Properly formatted	Record of insurance	Dental	No SSA	Not Disabled	Working
6	Male	Primary	Partially paid	Improperly formatted	Record of insurance	Medical	SSA	Disabled	Not Working
7	Male	Secondary	Partially paid	Properly formatted	No record of insurance	Dental	SSA	Disabled	Not Working
8	Male	Primary	None paid	Properly formatted	No record of insurance	Medical	No SSA	Disabled	Not Working
9	Female	Secondary	None paid	Improperly formatted	Record of insurance	Dental	SSA	Not Disabled	Not Working
10	<i>Female</i>	<i>Primary</i>	Fully paid	Improperly formatted	<i>No record of insurance</i>	Medical	<i>SSA</i>	Not Disabled	Working

6. Test Case Coverage Analysis



ANALYSIS

Results from Projects Using CTD

Project	Original Test Case Count	CTD Test Case Count	Results & Impact
A	667	230	Identified 23 missing tests and reduced number of test cases by 65%
B	27	112	Original test suite only had ~30% test coverage of defined test requirements. The increase of test cases closed our testing gaps.
C	228	17	Reduced number of test cases by 92%
D	121	84	Reduced number of test cases by 30%
E	215	120	Reduced number of test cases by 44%
F	61	41	Reduced number of test cases by 33%
G	34	52	Original test suite only had 65% test coverage of defined test requirements. The increase of test cases closed our testing gaps.

CONCLUSION

Conclusion

CTD is a great test planning process to supplement your testing strategy

CTD is an easy process which can be quickly adopted by any tester

It can help you save time and money in UAT by reducing the test planning phase along with the number of manual tests that need to be executed



QUESTIONS?