Software Testing Help

Software Testing Complete Guide

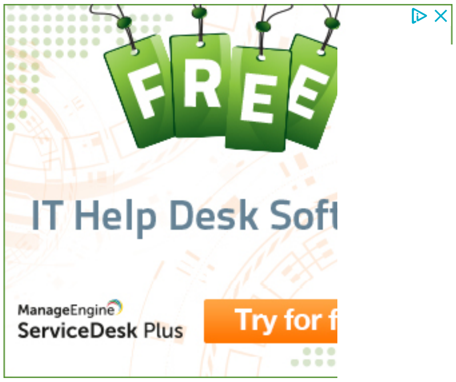
How to Dramatically Reduce the Number of Test Cases Combinations Using Pairwise Testing

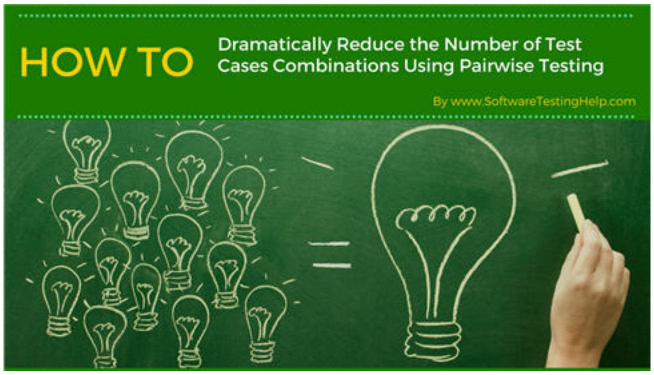
Posted In | Testing Methodologies, Types of Testing | Last Updated: "February 22, 2018"

What is Pairwise Testing and How It is Effective Test Design Technique for Finding Defects:

In this article, we are going to learn about a '**Combinatorial Testing**' technique called '**Pairwise Testing'** also known as '**All-Pairs Testing**'.

Smart testing is the need of the hour. 90% of the time's system testing team has to work with tight schedules. So test design techniques should be very effective for maximum test coverage and high defect yield rate.




HOW TO
Dramatically Reduce the Number of Test Cases Combinations Using Pairwise Testing
By www.SoftwareTestingHelp.com

Pairwise Testing is a test design technique that delivers hundred percent test coverage.

The output of a software application depends on many factors e.g. input parameters, state variables and environment configurations. Techniques like boundary value analysis and equivalence partitioning can be useful to identify the possible values for individual factors. But it is impractical to test all possible combinations of values for all those factors. So instead **a subset of combinations is generated** to satisfy all factors.

What You Will Learn: [show]

Definition: What is Pairwise Testing?

ISTQB defines All-Pairs Testing (or Pairwise Testing) as A black-box test design technique in which test cases are designed to execute all possible discrete combinations of each pair of input parameters.

All-Pairs technique is very helpful for designing tests for applications involving multiple parameters. Tests are designed such that for each pair of input parameters to a system, there are all possible discrete combinations of those parameters. The test suite covers all combinations; therefore it is not exhaustive yet very effective in finding bugs.

Let us learn how to apply* Pairwise Testing *by this example.

Pairwise Testing Example

Car Ordering Application:

- The car ordering application allows for Buying and Selling cars. It should support trading in Delhi and Mumbai.
- The application should have registration numbers, may be valid or invalid. It should allow the trade of following cars: BMW, Audi, and Mercedes.
- Two types of booking can be done: E-booking and In Store.
- Orders can be placed only during trading hours.

Step #1: Let's list down the variables involved.

1) Order category
a. Buy
b. Sell

2) Location
a. Delhi
b. Mumbai

3) Car brand
a. BMW
b. Audi
c. Mercedes

4) Registration numbers
a. Valid (5000)
b. Invalid

5) Order type
a. E-Booking
b. In-store

6) Order time
a. Working hours
b. Non-working hours

If we want to test all possible valid combinations:
= 2 X 2 X 3 X 5000 X 2 X 2
= 240000 Valid test cases combinations :(

There is also an infinite number of invalid combinations.

Step #2: Let's simplify

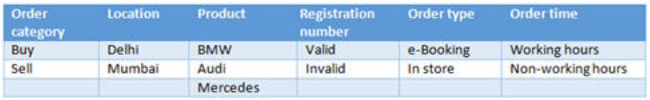
– Use a smart representative sample.
– Use groups and boundaries, even when data is non-discrete.
– Reduce Registration Number to Two

1. Valid registration number
2. Invalid registration number

Now let's calculate number of possible combinations
= 2 X 2 X 3 X 2 X 2 X 2
= 96

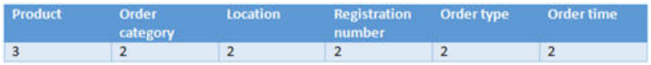
Step #3: Arranging variables and values involved.

When we arrange variables and values involved, it looks something like this.



Order category	Location	Product	Registration number	Order type	Order time
Buy	Delhi	BMW	Valid	e-Booking	Working hours
Sell	Mumbai	Audi	Invalid	In store	Non-working hours
		Mercedes			

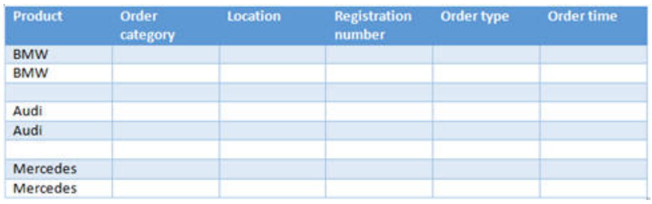
Now order the variables so that the one with the most number of values is first and the least is last.



Product	Order category	Location	Registration number	Order type	Order time
3	2	2	2	2	2

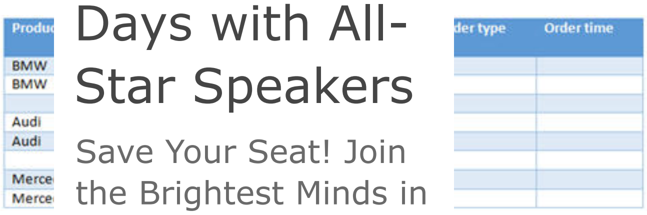
Step #4: Arrange variables to create test suite

Let's start filling in the table column by column. Initially, table should look something like this. The three values of **Product** (variable having the highest number of values) should be written two times each (two is the number of values of next highest variable i.e. **Order category**).



Product	Order category	Location	Registration number	Order type	Order time
BMW					
BMW					
Audi					
Audi					
Mercedes					
Mercedes					

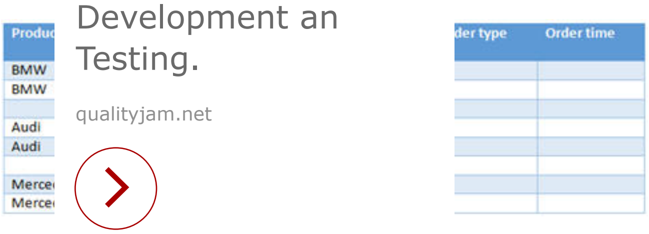
The Order Category column has two values. That's how many times we need to insert the values of the first column, Product.



Produc	der type	Order time
BMW		
BMW		
Audi		
Audi		
Merce		
Merce		

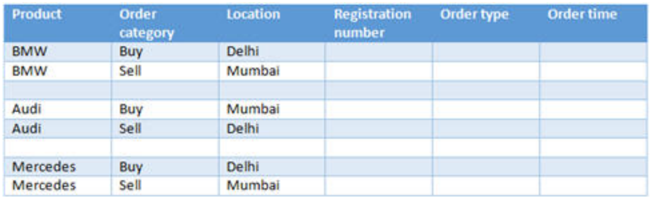
For ea

s of column 2. Repeat the same for column 3.



Produc	der type	Order time
BMW		
BMW		
Audi		
Audi		
Merce		
Merce		

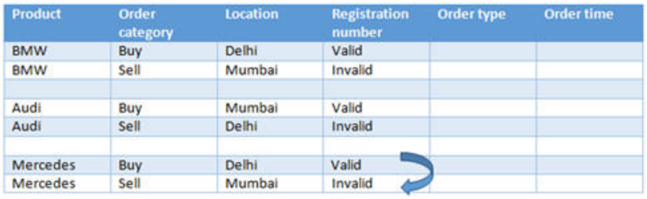
We ha nd Mumbai. We have a Sell and Mumbai, but
there's es in the second set in the third column.



Product	Order category	Location	Registration number	Order type	Order time
BMW	Buy	Delhi			
BMW	Sell	Mumbai			
Audi	Buy	Mumbai			
Audi	Sell	Delhi			
Mercedes	Buy	Delhi			
Mercedes	Sell	Mumbai			

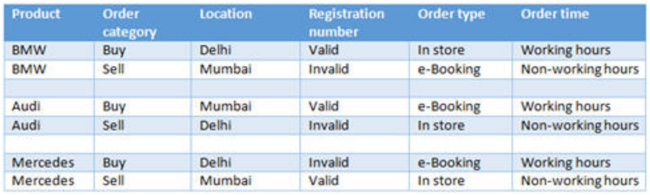
This looks much better!

We will repeat the same steps for column 3 and 4.



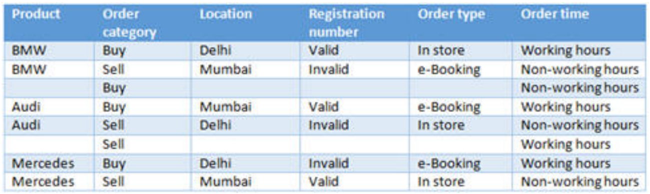
Product	Order category	Location	Registration number	Order type	Order time
BMW	Buy	Delhi	Valid		
BMW	Sell	Mumbai	Invalid		
Audi	Buy	Mumbai	Valid		
Audi	Sell	Delhi	Invalid		
Mercedes	Buy	Delhi	Valid		
Mercedes	Sell	Mumbai	Invalid		

When columns 3 and 4 are compared, each value in column 3 has both the values of column 4. But when you compare the 2nd and 4th column, we have Buy and Valid & Sell and Invalid .i.e. Buy does not have 'Invalid' and Sell does not have 'Valid'. Hence we need to interchange the last set of values in the 4th column.



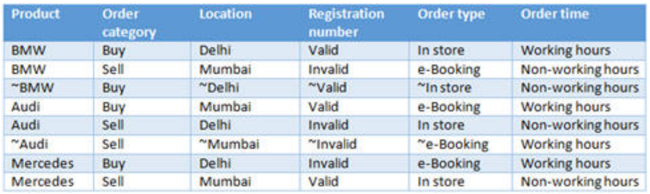
Product	Order category	Location	Registration number	Order type	Order time
BMW	Buy	Delhi	Valid	In store	Working hours
BMW	Sell	Mumbai	Invalid	e-Booking	Non-working hours
Audi	Buy	Mumbai	Valid	e-Booking	Working hours
Audi	Sell	Delhi	Invalid	In store	Non-working hours
Mercedes	Buy	Delhi	Invalid	e-Booking	Working hours
Mercedes	Sell	Mumbai	Valid	In store	Non-working hours

Column 6 (Order time) is problematic. We are missing Buy/Non-working hours and Sell/Working hours. We can't fit our missing pairs by swapping around values as we already swapped all the rows if we swap now we may miss other possible pairs which are already sorted. So, we add two more test cases that contain these pairs. Hence, the blank rows!



Product	Order category	Location	Registration number	Order type	Order time
BMW	Buy	Delhi	Valid	In store	Working hours
BMW	Sell	Mumbai	Invalid	e-Booking	Non-working hours
	Buy				Non-working hours
Audi	Buy	Mumbai	Valid	e-Booking	Working hours
Audi	Sell	Delhi	Invalid	In store	Non-working hours
	Sell				Working hours
Mercedes	Buy	Delhi	Invalid	e-Booking	Working hours
Mercedes	Sell	Mumbai	Valid	In store	Non-working hours

Now we will fill in the empty cells as we desire because the other variable values are purely arbitrary (or Don't Cares ~).



Product	Order category	Location	Registration number	Order type	Order time
BMW	Buy	Delhi	Valid	In store	Working hours
BMW	Sell	Mumbai	Invalid	e-Booking	Non-working hours
~BMW	Buy	~Delhi	~Valid	~In store	Non-working hours
Audi	Buy	Mumbai	Valid	e-Booking	Working hours
Audi	Sell	Delhi	Invalid	In store	Non-working hours
~Audi	Sell	~Mumbai	~Invalid	~e-Booking	Working hours
Mercedes	Buy	Delhi	Invalid	e-Booking	Working hours
Mercedes	Sell	Mumbai	Valid	In store	Non-working hours

Hurray! All pairs in 8 cases, instead of all combinations in 96!

Hence, we saw how efficient All-pairs technique of test design is. There stands a good chance of finding bugs and it is fun and powerful.

The pairwise testing technique has some limitations as well.

- It fails when the values selected for testing are incorrect.
- It fails when highly probable combinations get too little attention.
- It fails when interactions between the variables are not understood well.

Pairwise Testing Tools:

Tools are available that applies the all-pairs testing technique that facilitates us to effectively automate the Test Case Design process by generating a compact set of parameter value choices as the desired Test Cases. Some well-known tools from the industry are:

- PICT – 'Pairwise Independent Combinatorial Testing', provided by Microsoft Corp.
- IBM FoCuS – 'Functional Coverage Unified Solution', provided by IBM.
- ACTS – 'Advanced Combinatorial Testing System', provided by NIST, an agency of the US Government.
- Hexawise
- Jenny
- Pairwise by Inductive AS
- VPTag free Pairwise Testing Tool

Conclusion:

The pairwise testing technique can dramatically reduce the number of combinations to be covered but remains very effective in terms of fault detection. It is indeed a smart test design technique that guarantees a win-win situation for both test effort and test effectiveness.

During the Test planning phase of software testing, Pairwise testing technique should always be taken into consideration. Either we are doing it manually or using any tool to generate test cases, it becomes a necessary component of the test plan because it, in turn, affects Test estimation.

Share 94 Tweet +1 16 Pin 3 Share 140

← PREV POST | NEXT POST →

14 comments ↓

#1 **Aniket**

kudos for the great example. do you have more examples to share?

#2 **Meg**

Clear explanation on how pairwise combinations are created. Thanks!

#3 **Shraddha**

Gr8 explanation and nice information
Thanks!

#4 **Sunil**

Nice explanation. Another example would be good!

#5 **Vinay HN**

Nice and clear explanation.. Thumbs up..:)

#6 **Binu**

Really has nice concept reducing the test cases….Next Pairwise.

#7 **Gaurav Khurana**

Good to know that pairwise testing is so much important that few companies have created tool for it.

what kind of things these tools have provided apart from combination

#8 **Nikolay Advolodkin**

Really useful article. I enjoyed the read :)

#9 **Prathiba**

Nice article. Can you explain anyone of the pairwise testing tool widely using across the industry.

#10 **Peter**

If you swap buy/sell in 2nd column, you will not to add rows in step 6, right? So why don't you do that?

#11 **Vamsi**

Simply and clear, It explains the concept in deep. Thank you:)

#12 **Ashwini**

thanks for this explanation, its useful.
what is concept of 1-wise and 2-wise pair testing? how this will be calculated ?

#13 **Sam**

Why don't we have 9th test case for Buy – invalid – non working hours

#14 **Rupali**

Hi,

i am unable to download Advanced Combinatorial Testing System (ACTS) tool. Can some one help me out ? Actually i am looking for pairwise testing open source tool

Thanks and Regrads,
Rupali.

Leave a Comment

______ **Name***

______ **Email***

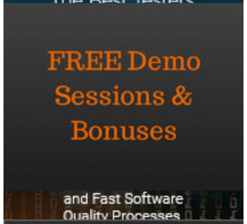
______ **Website (Optional)**

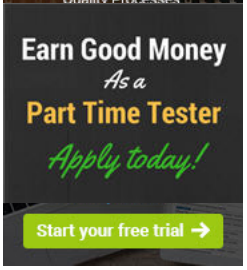
Submit

- ## Related Articles
  - What is Boundary value analysis and Equivalence partitioning?
  - What is Orthogonal Array Testing Technique (OATS) Technique?
  - Black Box Testing: Types and techniques of BBT
  - How to Write Complex Business Logic Test Scenarios Using Decision Table Technique
  - What is Error Guessing Technique?
  - Some tricky question answers
  - ISTQB Exam Questions on Equivalence partitioning and Boundary Value Analysis
  - Sample Test Cases for Testing Graphs and Charts
  - What is Defect Based Testing Technique?
  - ISTQB Foundation level exam Sample Paper – II

- ## Help & Tutorials
  - ISTQB Study Guide
  - ISTQB Premium Study Guide
  - Free QA Training
  - Free Selenium Training
  - Free QTP Training
  - Free QC Training
  - HP LoadRunner Tutorials
  - Free JMeter Tutorials
  - JIRA Tutorials
  - VBScript Tutorials
  - 101+ Interview Questions
- ## Adv

- ## Online Training by STH


FREE Demo Sessions & Bonuses
and Fast Software Quality Processes



Search

About us | Contact us | Advertise | Testing Services

All articles are copyrighted and can not be reproduced without permission.

Thanks For Reading! 50 Million Visits and Counting!