XP CONFERENCE INDIA 2016

Powered By SolutionsIQ

19 - 20 August 2016
Bangalore

www.xpconference.in

Pairwise Testing in Agile World

Classic Combinatorial Test case design technique that can be applied to agile testing to save testing cost & time.

How many combinations does it take to test this?

Four fundamental challenges of software testing
Source : Cem Kaner

- Complete testing is impossible
- Testers misallocate resources because they fall for the company's process myths
- Test groups operate under multiple missions, often conflicting, rarely articulated
- Test groups often lack skilled programmers, and a vision of appropriate projects that would keep programming testers challenged

Complete testing is IMPOSSIBLE (Cem Kaner)

- Test every possible input to every variable
- Test every possible combination of inputs to every combination of variables
- Test every possible sequence through the program
- Test every hardware / software configuration, including configurations of servers not under your control
- Test every way in which the user might try to use the program

Motivation for Pairwise Testing

- Wallace and Kuhn of NIST (National Institute of Standards & Technology) determined that 98% of the reported software defects in recalled medical devices could have been detected by testing all pairs of parameter settings.

D.R. Wallace, D.R. Kuhn, “Failure Modes in Medical Device Software: An Analysis of 15 Years of Recall Data,” Intl. Journal of Reliability, Quality, and Safety Engineering, vol. 8, no. 4.

Essence of Pairwise Testing

Parameter Name	Values	No of Values
Age	10, 14, 30, 50	4
Gender	Male, Female	2
Certainty	Probable, Doubtful, Possible, Established, Suspected	5
Severity	Minor, Moderate, Major	3
Food	True, False	2
Ethanol	True, False	2
Medication	COUMADIN 10 MG ORAL TABS, ASPIRIN 325 MG ORAL TABS, STENDRA TABS, ABILIFY TABS	4

Total Possible Configurations:

4 x 2 x 5 x 3 x 2 x 2 x 4 = 1920

Software Failure Analysis

- The simplest bugs in a system are generally triggered by a single input parameter

  E.g., failure occurs if

  - pressure < 10 (1-way interaction)

- The next simplest category of bugs consists of those dependent on interactions between pairs of parameters

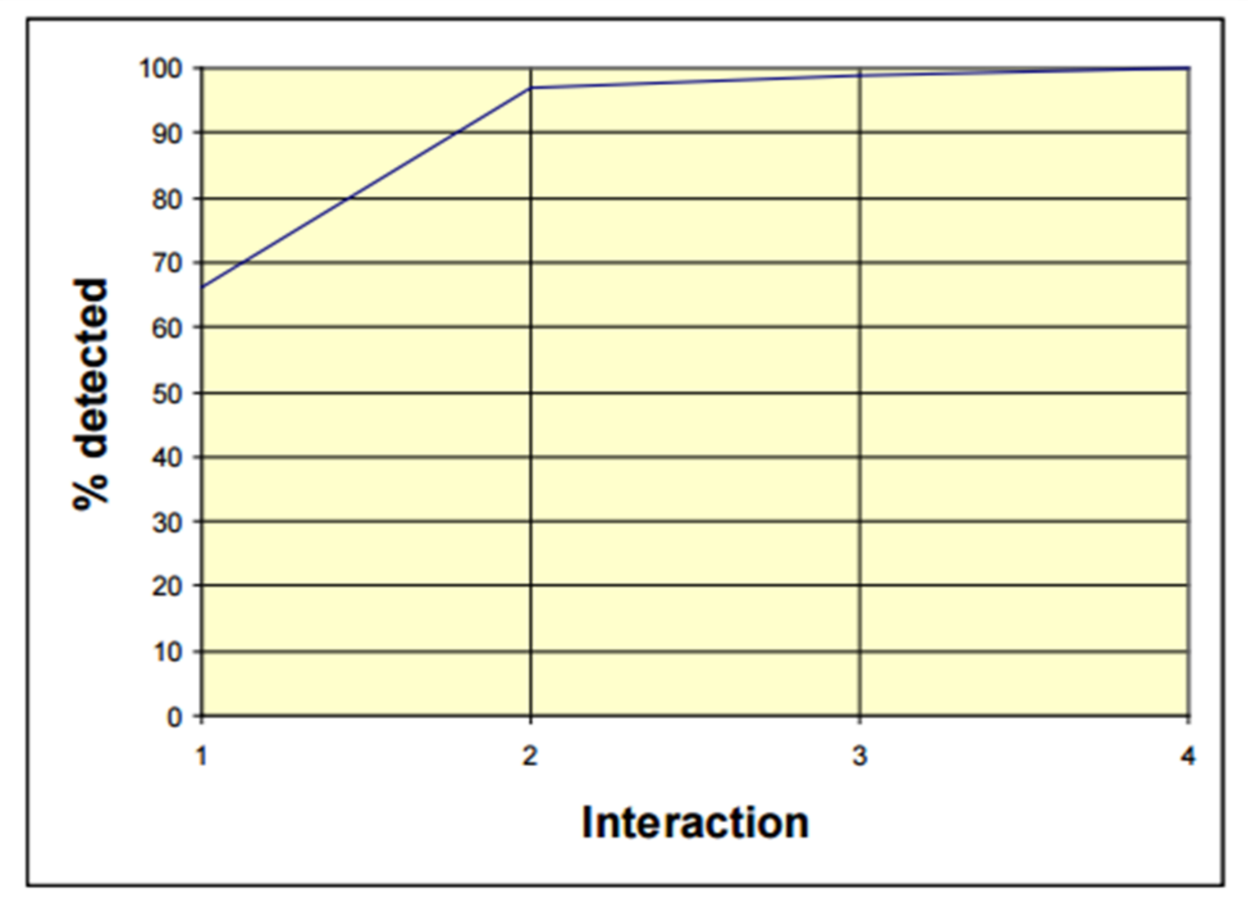
  E.g., failure occurs if

  - pressure < 10 & volume > 300 (2-way interaction)

Software Failure Analysis

- Bugs involving interactions between 3 or more parameters are progressively less common, while at the same time progressively more expensive
- Most complex failure required 4-way interaction

Software Failure Analysis – Medical Devices

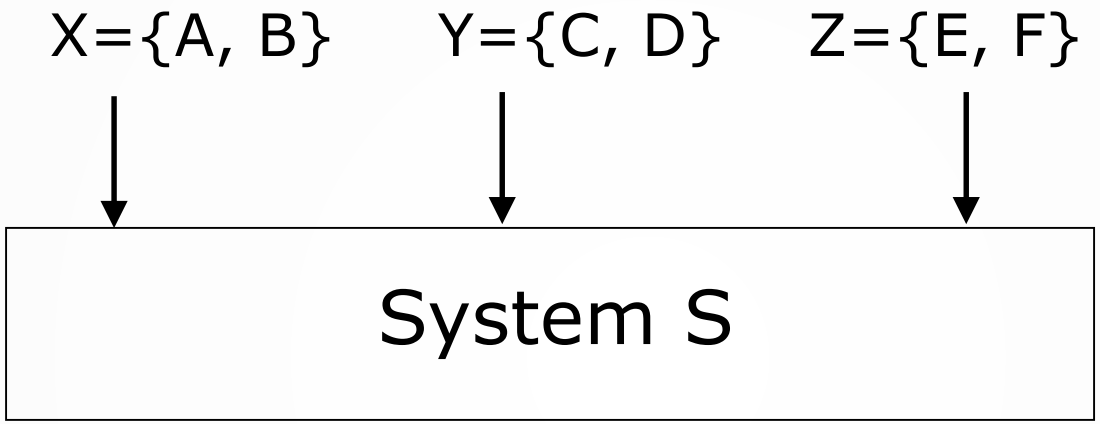


The Pairwise Technique

Don’t test all combinations. Test all pairs.

- Pairwise testing – an approach to combinatorial testing that executes a pairwise test data set.
- Pairwise test data set - A set of test cases that covers all combinations of the selected test data values for every pair of a system’s input variables.
- Pairwise testing is also known as 2-way testing; other approaches include 3-way, 4-way, … , n-way.

An Example



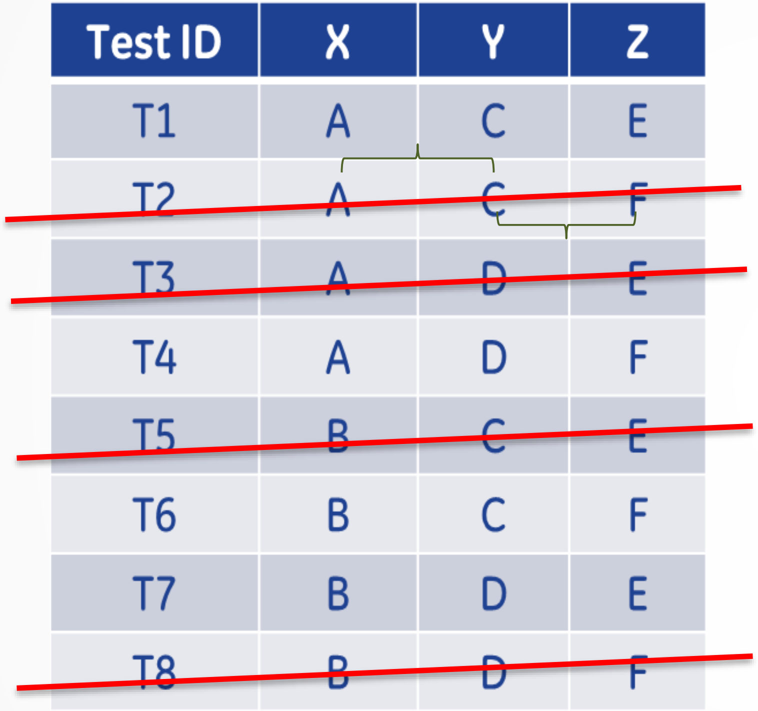
There are 2^3 = 8 Combinations

All Combinations (Test Cases)

X={A, B}   Y={C, D}   Z={E, F}

Test ID	X	Y	Z
T1	A	C	E
T2	A	C	F
T3	A	D	E
T4	A	D	F
T5	B	C	E
T6	B	C	F
T7	B	D	E
T8	B	D	F

Pairwise Reductions by hand



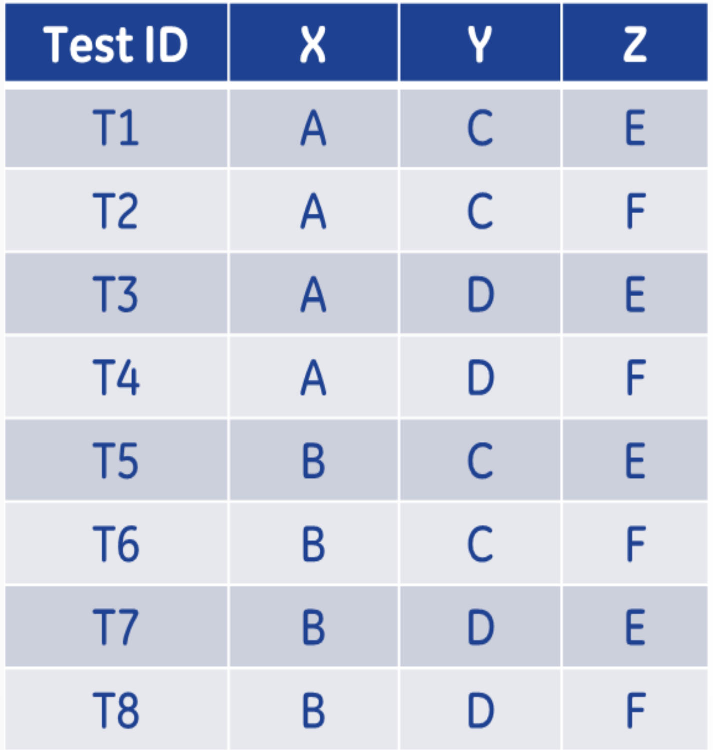
Test ID	X	Y	Z
T1	A	C	E
~~T2~~	~~A~~	~~C~~	~~F~~
~~T3~~	~~A~~	~~D~~	~~E~~
T4	A	D	F
~~T5~~	~~B~~	~~C~~	~~E~~
T6	B	C	F
T7	B	D	E
~~T8~~	~~B~~	~~D~~	~~F~~

You can get rid of test case **T2**, since

- AC is covered in T1
- CF is covered in T6
- AF is covered in T4

What other test cases can you get rid of?

Pairwise Reductions by hand



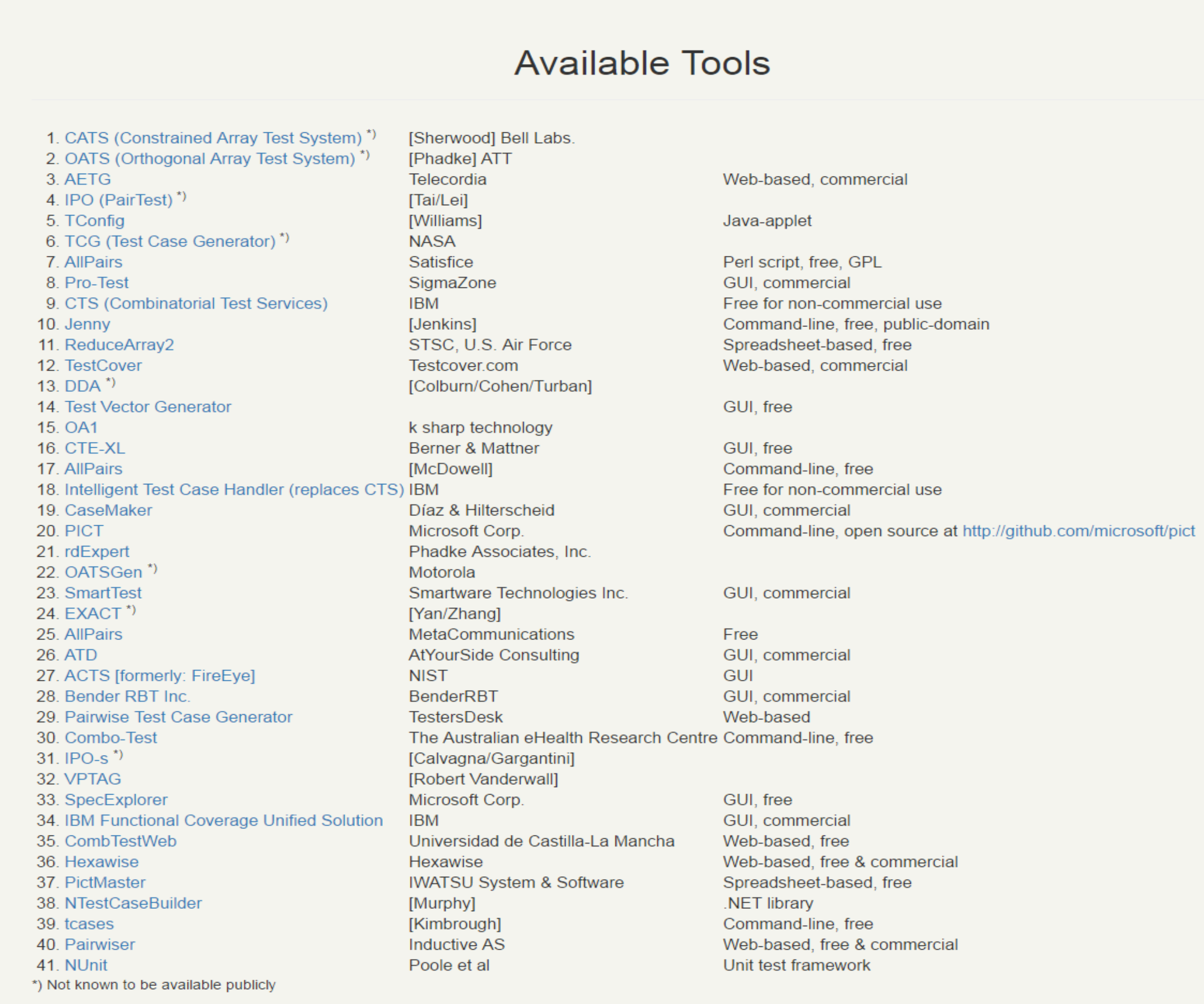
Test ID	X	Y	Z
T1	A	C	E
T2	A	C	F
T3	A	D	E
T4	A	D	F
T5	B	C	E
T6	B	C	F
T7	B	D	E
T8	B	D	F

Test ID	X	Y	Z
T1	A	C	E
T4	A	D	F
T6	B	C	F
T7	B	D	E

Pairwise Reductions

Number of inputs	Number of selected test data values	Total Number of combinations	Size of pairwise test set
3	2	8	4
7	2	128	8
13	3	1594323	18
40	3	1.2 x 10^{19}	21

Pairwise Reduction - Tools



Available Tools

1.	CATS (Constrained Array Test System) *)	[Sherwood] Bell Labs.	
2.	OATS (Orthogonal Array Test System) *)	[Phadke] ATT	
3.	AETG	Telecordia	Web-based, commercial
4.	IPO (PairTest) *)	[Tai/Lei]	
5.	TConfig	[Williams]	Java-applet
6.	TCG (Test Case Generator) *)	NASA	
7.	AllPairs	Satisfice	Perl script, free, GPL
8.	Pro-Test	SigmaZone	GUI, commercial
9.	CTS (Combinatorial Test Services)	IBM	Free for non-commercial use
10.	Jenny	[Jenkins]	Command-line, free, public-domain
11.	ReduceArray2	STSC, U.S. Air Force	Spreadsheet-based, free
12.	TestCover	Testcover.com	Web-based, commercial
13.	DDA *)	[Colburn/Cohen/Turban]	
14.	Test Vector Generator		GUI, free
15.	OA1	k sharp technology	
16.	CTE-XL	Berner & Mattner	GUI, free
17.	AllPairs	[McDowell]	Command-line, free
18.	Intelligent Test Case Handler (replaces CTS)	IBM	Free for non-commercial use
19.	CaseMaker	Díaz & Hilterscheid	GUI, commercial
20.	PICT	Microsoft Corp.	Command-line, open source at http://github.com/microsoft/pict
21.	rdExpert	Phadke Associates, Inc.	
22.	OATSGen *)	Motorola	
23.	SmartTest	Smartware Technologies Inc.	GUI, commercial
24.	EXACT *)	[Yan/Zhang]	
25.	AllPairs	MetaCommunications	Free
26.	ATD	AtYourSide Consulting	GUI, commercial
27.	ACTS [formerly: FireEye]	NIST	GUI
28.	Bender RBT Inc.	BenderRBT	GUI, commercial
29.	Pairwise Test Case Generator	TestersDesk	Web-based
30.	Combo-Test	The Australian eHealth Research Centre	Command-line, free
31.	IPO-s *)	[Calvagna/Gargantini]	
32.	VPTAG	[Robert Vanderwall]	
33.	SpecExplorer	Microsoft Corp.	GUI, free
34.	IBM Functional Coverage Unified Solution	IBM	GUI, commercial
35.	CombTestWeb	Universidad de Castilla-La Mancha	Web-based, free
36.	Hexawise	Hexawise	Web-based, free & commercial
37.	PictMaster	IWATSU System & Software	Spreadsheet-based, free
38.	NTestCaseBuilder	[Murphy]	.NET library
39.	tcases	[Kimbrough]	Command-line, free
40.	Pairwiser	Inductive AS	Web-based, free & commercial
41.	NUnit	Poole et al	Unit test framework

*) Not known to be available publicly

TestCover

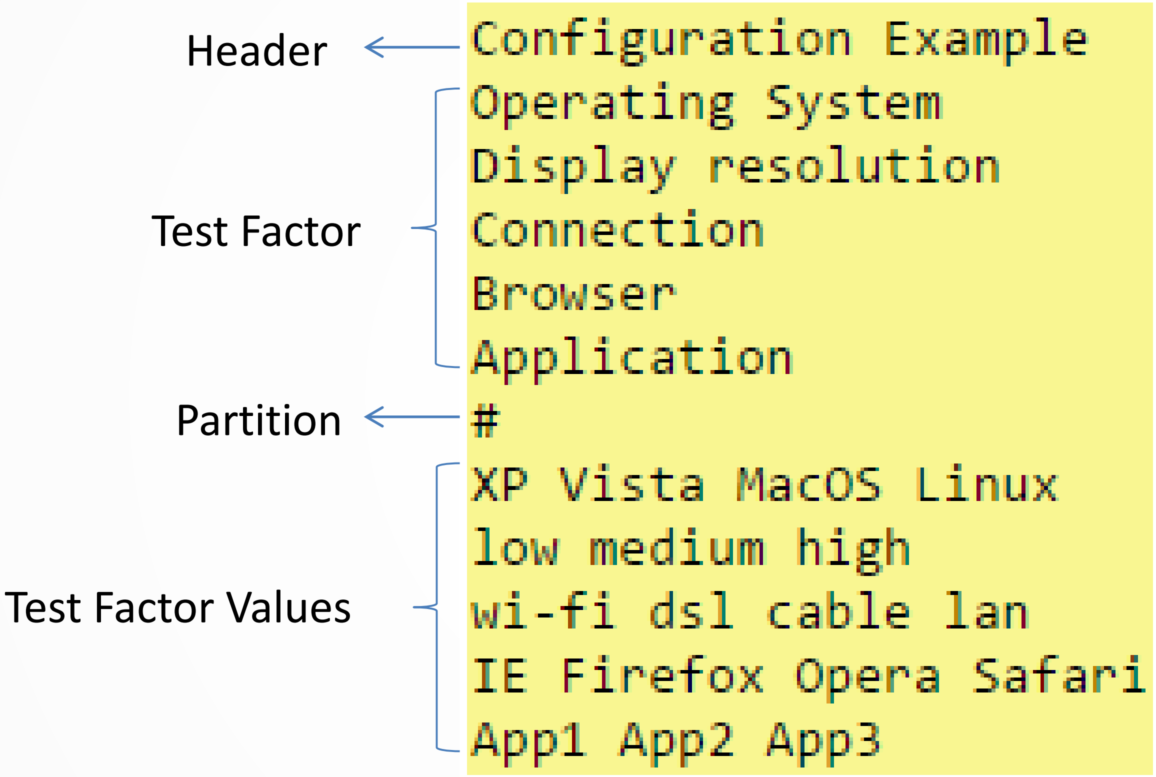
- Commercial off the shelf tool
  - Algorithm based on Bell Labs research
  - Mathematically optimal among other tools
- Determines test scenarios/cases based on input
  - Test Factor
  - Test Factor Values
  - Documented Constraints
- https://testcover.com/sub/index.php

TestCover Example

Test Factor	Number of Values	Test Factor Values
Operating system	4	XP, Vista, MacOS, Linux
Display resolution	3	Low, medium, high
Connection	4	wi-fi, dsl, cable, lan
Browser	4	IE, Firefox, Opera, Safari
Application	3	App1, App2, App3

Total Configurations = **576**

The TestCover Request



Header

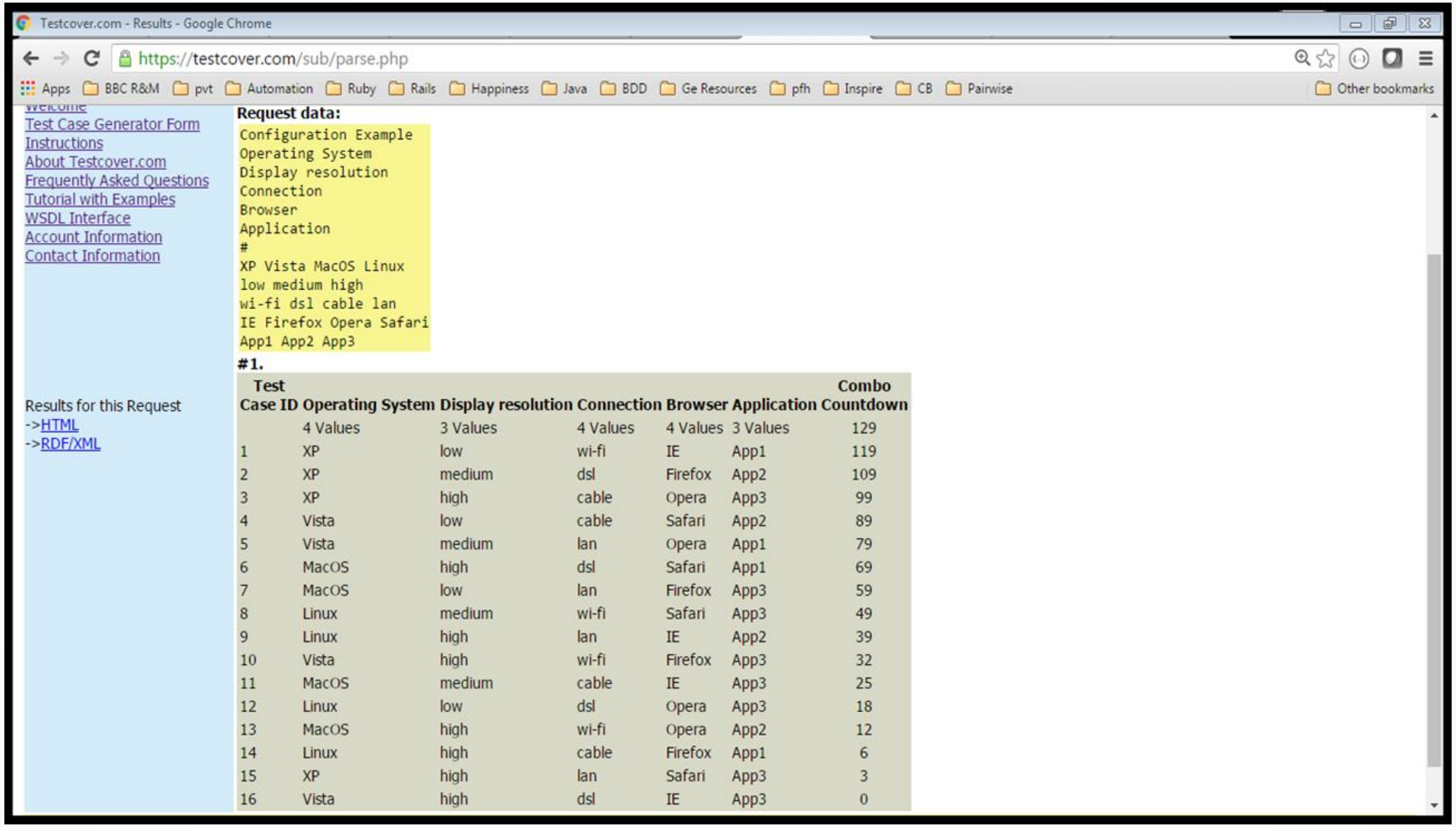
Test Factor

Partition

Test Factor Values

```
Configuration Example
Operating System
Display resolution
Connection
Browser
Application
#
XP Vista MacOS Linux
low medium high
wi-fi dsl cable lan
IE Firefox Opera Safari
App1 App2 App3
```

The Reduced Pairwise Subset



Welcome
Test Case Generator Form
Instructions
About Testcover.com
Frequently Asked Questions
Tutorial with Examples
WSDL Interface
Account Information
Contact Information

Results for this Request
->HTML
->RDF/XML

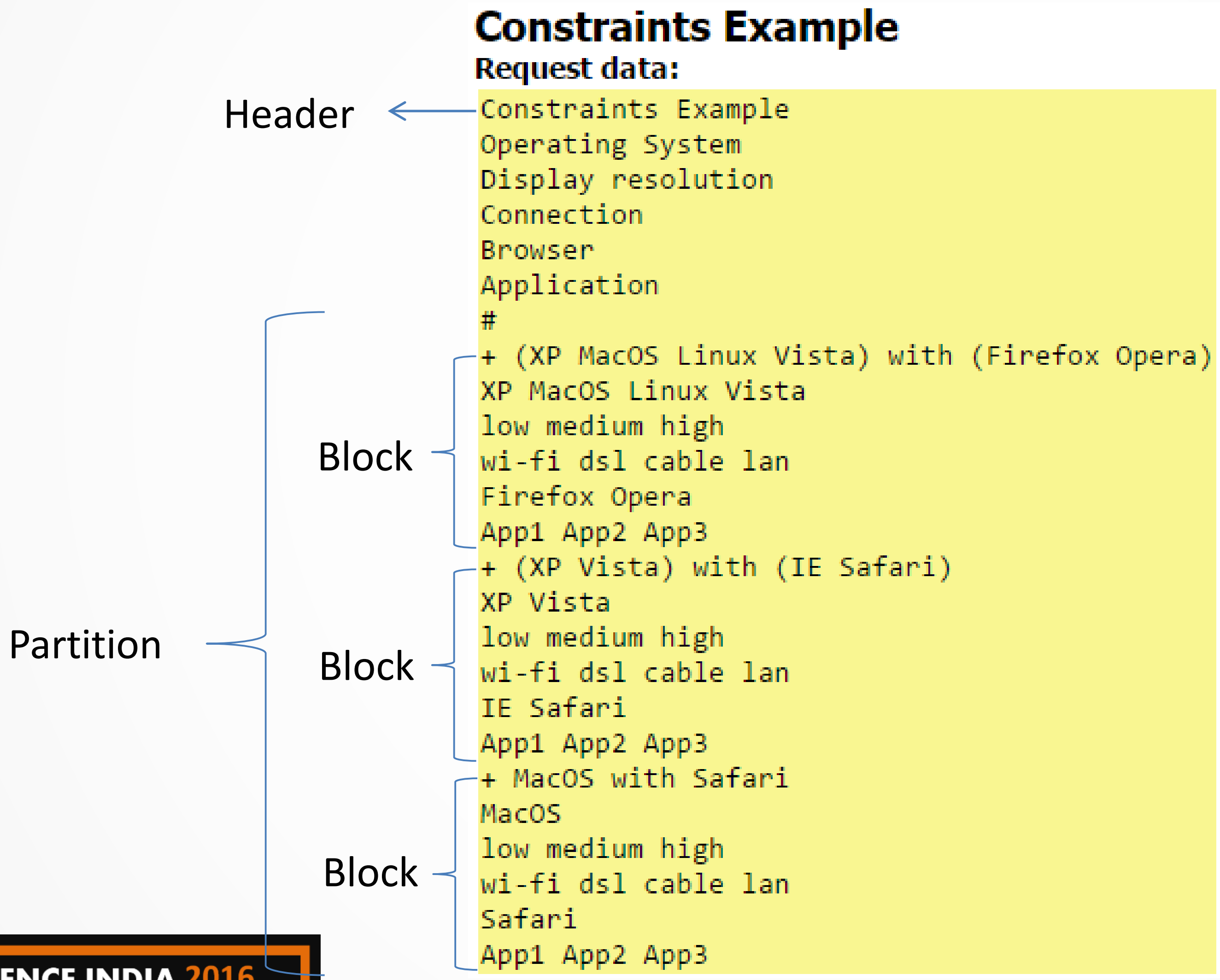
Request data:

```
Configuration Example
Operating System
Display resolution
Connection
Browser
Application
#
XP Vista MacOS Linux
low medium high
wi-fi dsl cable lan
IE Firefox Opera Safari
App1 App2 App3
```

#1.

Test Case ID	Operating System	Display resolution	Connection	Browser	Application	Combo Countdown
	4 Values	3 Values	4 Values	4 Values	3 Values	129
1	XP	low	wi-fi	IE	App1	119
2	XP	medium	dsl	Firefox	App2	109
3	XP	high	cable	Opera	App3	99
4	Vista	low	cable	Safari	App2	89
5	Vista	medium	lan	Opera	App1	79
6	MacOS	high	dsl	Safari	App1	69
7	MacOS	low	lan	Firefox	App3	59
8	Linux	medium	wi-fi	Safari	App3	49
9	Linux	high	lan	IE	App2	39
10	Vista	high	wi-fi	Firefox	App3	32
11	MacOS	medium	cable	IE	App3	25
12	Linux	low	dsl	Opera	App3	18
13	MacOS	high	wi-fi	Opera	App2	12
14	Linux	high	cable	Firefox	App1	6
15	XP	high	lan	Safari	App3	3
16	Vista	high	dsl	IE	App3	0

Another Example – with Constraints

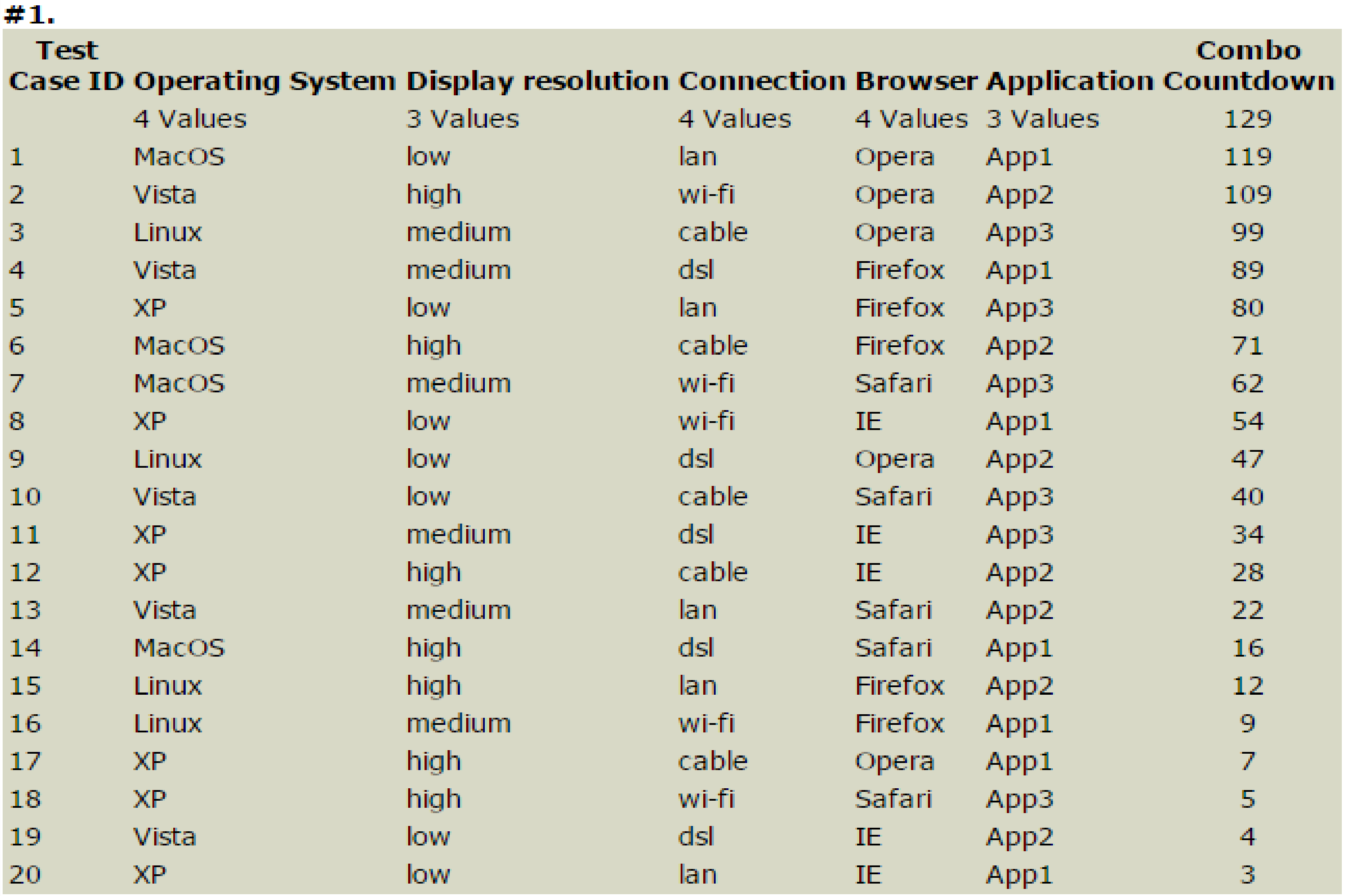


Constraints Example

Request data:

```
Constraints Example
Operating System
Display resolution
Connection
Browser
Application
#
+ (XP MacOS Linux Vista) with (Firefox Opera)
XP MacOS Linux Vista
low medium high
wi-fi dsl cable lan
Firefox Opera
App1 App2 App3
+ (XP Vista) with (IE Safari)
XP Vista
low medium high
wi-fi dsl cable lan
IE Safari
App1 App2 App3
+ MacOS with Safari
MacOS
low medium high
wi-fi dsl cable lan
Safari
App1 App2 App3
```

The Result



#1.

Test						**Combo**
Case ID	**Operating System**	**Display resolution**	**Connection**	**Browser**	**Application**	**Countdown**
	4 Values	3 Values	4 Values	4 Values	3 Values	129
1	MacOS	low	lan	Opera	App1	119
2	Vista	high	wi-fi	Opera	App2	109
3	Linux	medium	cable	Opera	App3	99
4	Vista	medium	dsl	Firefox	App1	89
5	XP	low	lan	Firefox	App3	80
6	MacOS	high	cable	Firefox	App2	71
7	MacOS	medium	wi-fi	Safari	App3	62
8	XP	low	wi-fi	IE	App1	54
9	Linux	low	dsl	Opera	App2	47
10	Vista	low	cable	Safari	App3	40
11	XP	medium	dsl	IE	App3	34
12	XP	high	cable	IE	App2	28
13	Vista	medium	lan	Safari	App2	22
14	MacOS	high	dsl	Safari	App1	16
15	Linux	high	lan	Firefox	App2	12
16	Linux	medium	wi-fi	Firefox	App1	9
17	XP	high	cable	Opera	App1	7
18	XP	high	wi-fi	Safari	App3	5
19	Vista	low	dsl	IE	App2	4
20	XP	low	lan	IE	App1	3

Caveat

- Pairwise testing will not be effective if you choose the wrong input test data values
- We often assume defects can be immediately observed. In fact, there may be slow internal corruption occurring
- High-use (including defaults) or high-risk combinations probably don't get enough attention

Note: These dangers are not exclusive to pairwise testing. They affect all testing efforts

References

- http://www.pairwise.org
- https://testcover.com/sub/index.php
- http://csrc.nist.gov/groups/SNS/acts/documents/verify07.pdf
- http://www.bcs.org/upload/pdf/lcopeland-070312.pdf