Testcover.com

Your test designs. Covered.™

Why do we test? We live in an age of wonders, with missions to Mars, self-driving cars, the Internet of Things, and systems both too big to test and too big to fail. Since the beginning of computing we've tested interactions: *this* thing with *that* thing, *that* thing with the *other* thing, *this* thing with the *other* thing… But there are way too many interactions, and they increase exponentially with the introduction of each new technology.

Combinatorial testing offers a way to address this challenge. Consider a system having 30 things that may interact (30 test factors). Suppose that 6 factors have 3 settings or values, and 24 factors have 2 values. There are $3^6\ 2^{24}$ = 12,230,590,464 possible test cases. But only 15 are needed to test all the interactions between pairs of values. Guessing and random selection of test cases typically do *not* cover all the interactions.

Simply the best Testcover.com delivers a simple, practical solution. We believe it is the best one. We will match it with any test case generator – commercial, research, government-issued, homegrown – *any*. Here's why.

Efficient test designs Testing software systems can be expensive; not testing is even more expensive. So software engineers need a *minimal* number of test cases to limit costs and meet schedules. Secondly, generating test cases must be quick to keep up with a professional development team. What-if designs to choose among test parameters are normal. And sometimes late changes to test plans are unavoidable.

The table shows the number of test cases generated and the time it takes for various examples run on our service. The test model for each example is in product notation. In the $3^6\ 2^{24}$ model, 15 test cases are generated in about 1 second. In several examples the number of test cases generated is the smallest possible result.

Testcover.com performance		
Test model [values^{factors}]	**Test cases**	**Response time [mm:ss]**
2^{120}	10	00:02
$3^6\ 2^{24}$	15	00:01
3^{90}	23	00:02
$4^1\ 3^{39}\ 2^{35}$	21	00:02
$4^{15}\ 3^{17}\ 2^{29}$	29	00:02
$5^2\ 3^9\ 2^{88}$	25	00:03
$6^3\ 3^{12}$	36	00:02
7^{63}	91	00:06
$8^6\ 5^{10}\ 3^{83}$	64	00:05
9^{90}	153	00:26
$10^3\ 4^{96}$	100	00:08
$15^2\ 9^6\ 8^6\ 7^6\ 6^6\ 5^6\ 4^6\ 3^6\ 2^6$	225	00:16
$20^1\ 19^1\ 18^2\ 15^3\ 12^3\ 8^5\ 5^{10}\ 4^{10}\ 3^{15}$	380	00:43
$25^3\ 24^4\ 20^5\ 16^6\ 12^6\ 8^8\ 4^8\ 3^{10}$	625	04:18
$30^1\ 27^3\ 21^5\ 17^5\ 14^5\ 12^5\ 5^5\ 4^5\ 3^6\ 2^{10}$	810	06:02

Effective test designs Test plans must cover all the interactions needed to meet test objectives. These may include test factors for configuration options, user inputs, database records, test automation choices, system states, and more. The service supports all of these by generating *covering arrays*, templates that insure required interactions are included.

It is just as important *not* to include interactions that are unsupported or impossible. A test case that calls for running the Internet Explorer (IE) browser on the Linux operating system is impossible. Yet you cannot simply omit such test cases because they may also contain other, *required* interactions. Testcover.com covers all allowed interactions while avoiding invalid combinations. The service handles constraints two ways, with simple, *basic blocks* and with *embedded functions*.

Easy usability Generator requests are easy with Testcover.com's Direct Product Block (DPB) notation. Values for each test factor are listed on a separate line in a block. All combinations in a block are allowed. With Basic Blocks, constraints are handled with multiple blocks. These are indicated by lines starting with a plus (+). Combinations from *all* the blocks are allowed, and combinations that don't appear in any blocks are disallowed.

In the browser constraint example two blocks are used, one for Windows and one for Linux. Three factors are included: OS, Browser and Application. Their values are listed on corresponding lines in the two blocks. Here's the request.

```
Browser constraint with Basic Blocks
OS
Browser
Application
#
+ Windows block
Windows
IE Firefox Chrome
App1 App2 App3
+ Linux block
Linux
Firefox Chrome
App1 App2 App3
```

For this request the service generates 9 test cases (the minimum number) and covers 20 of the 21 interactions. There is no test case for Linux with IE.

Embedded functions technology Embedded functions is a new Testcover.com feature, which was demonstrated at IWCT 2016 and deployed in June 2017. It simplifies the specification of constraints even further. In the example the browser values are determined by the operating system factor (OS). So the allowed browser values can be expressed by this simple function written in PHP.

```
function fBrowser($OS) {
  global $sp;          /* separator character */
  switch($OS) {
    case 'Windows':
      $Browser='IE'.$sp.'Firefox'.$sp.'Chrome';
      break;
    case 'Linux':
      $Browser='Firefox'.$sp.'Chrome';
      break;
  }
  if(isset($Browser)) return($Browser);
}
```

The generator request has only one block now, with the browser values given by the fBrowser($OS) function. (The OS factor becomes $OS here because it represents a PHP variable.)

```
Browser constraint with fBrowser function
$OS
Browser
Application
#
Windows Linux
fBrowser($OS)
App1 App2 App3
```

This request generates 9 test cases and covers the same interactions as the previous one. Embedded functions are particularly useful when there are complex constraints which would require many blocks. And functions can be reused in different requests.

Test design automation Equivalence partitioning is a well known test design technique for covering classes of equivalent expected results. Typically engineers manually select combinations of test factors to reach each class once. With embedded functions the selection of test factor values can be automated, to save time and improve accuracy.

Based on requirements, equivalence class functions are written to return values for equivalence class factors. These factors indicate the expected results classes. In each test case the presence of a particular class insures a combination of its determinant factor values is there also. For more details on how this works, see Test design automation: equivalence classes, boundaries, edges and corner cases. Examples with specified test cases are included.

No IT required Testcover.com provides Software as a Service. Users need a standard browser. There is no need for new computers, installation, set-up or software updates. You can start using the service in minutes and take it wherever the internet goes. It can be used with a variety of development environments, processes and tools.

Group subscriptions We offer group subscriptions for software engineering organizations. Group subscriptions include an optional administration interface at no additional cost. The interface allows a designated user to perform several management tasks:

- List available subscription activation codes
- Register a new user – select an activation code and use forms prepopulated with group information
- List active users and their respective renewal dates
- Reset password for an active user
- Renew subscriptions for active users
- Suspend subscriptions and transfer remaining subscription time to new activation codes
- Purchase new subscriptions

Innovative leadership Our record is one of recognizing needs and providing solutions before others. Our founder built a combinatorial test case generator in 1990 to improve software test designs for a well-known commercial development organization. We believe it was the first use of a greedy search to accommodate real-world configuration constraints. Testcover.com has been providing its Software as a Service since 2003, when the Cloud was still blue sky. We have developed and introduced state-of-the-art array constructions and generation algorithms ever since. With embedded functions we provide a new technology offering significant software test design improvements in automation, accuracy, control and flexibility.

Try it now Get your activation code for a 30 day free trial at testcover.com/pub/prices.php. In just a few minutes you will see for yourself that Testcover.com is simply the best solution for covering test designs.