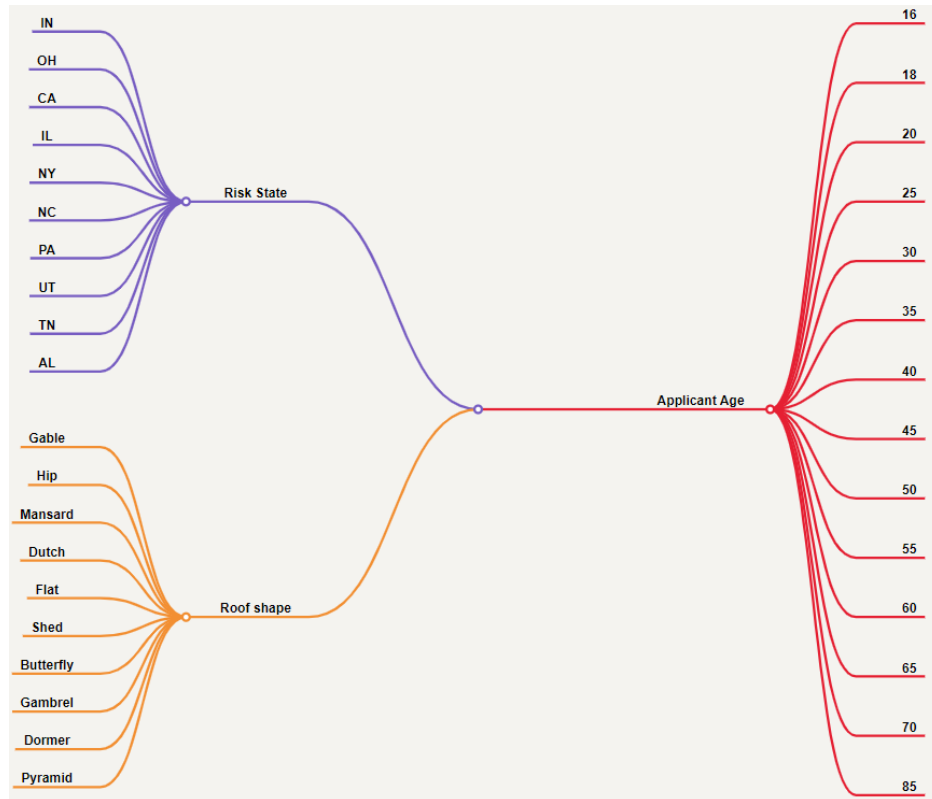


# Why “Test Case Count” Can Be a Misleading Metric in Model-Based Testing (And What To Do About It)

The default 2-way set of scenarios generated by the Test Case Designer is often a solid choice regarding the balance of quantity and quality. However, it may not be right for you, depending on the project scope, timelines, etc.

4 strategies can help you optimize the scenario suite further and tailor it to your goals. We will look at the “metamorphosis” of a single model and discuss those features in the increasing order of effort.

The base model:



The “benchmark” scenario count: **152** at 2-way strength.



We have chosen a small number of parameters to make it easier to track the impact. The steps below are applicable regardless of the model size, but the count reduction metric would of course depend on the model shape.

- Strategy 1 – Select a subset of tests
- Strategy 2 – Adjust the coverage dial setting away from 2-way
- Strategy 3 – Change how you describe your inputs (e.g., with optimizations to Parameters, Values, and/or Value Expansions)
- Strategy 4 – Add “artificial” constraints
- Conclusion

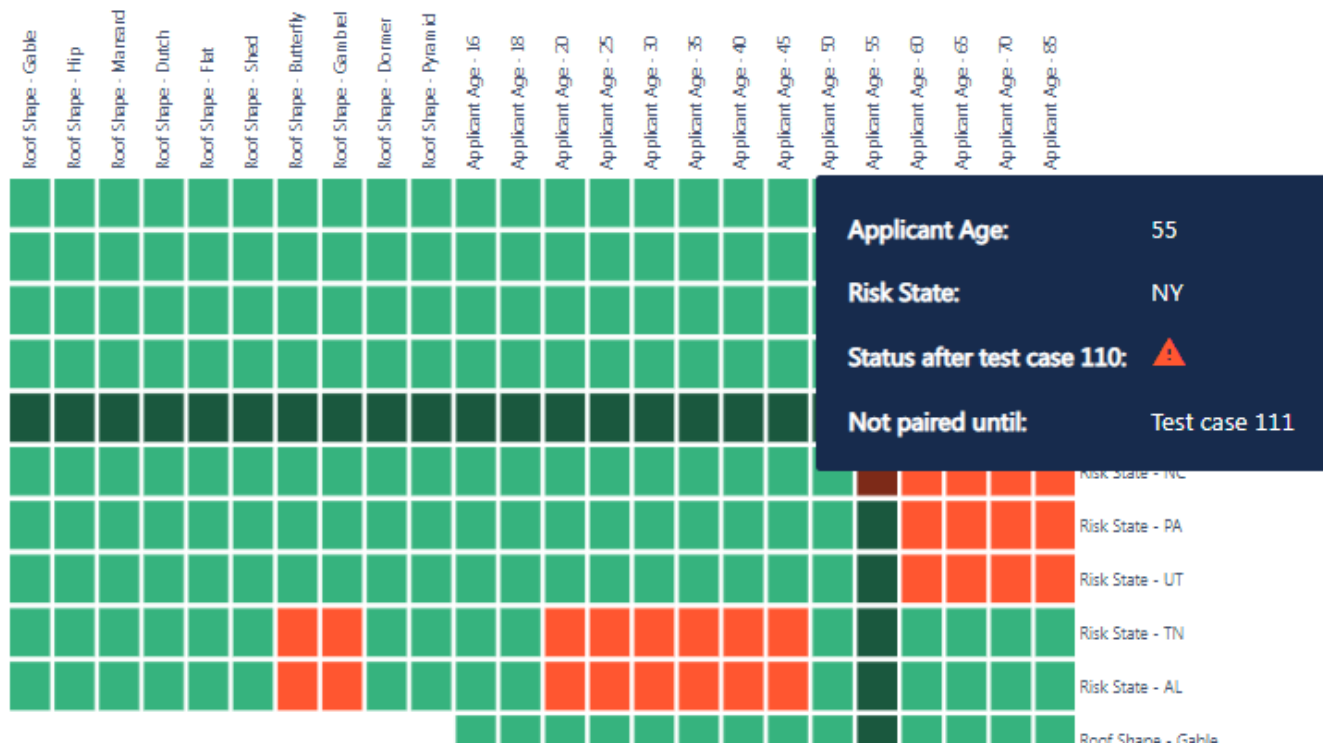
## Strategy 1 – Select a subset of tests

This method leverages the Analysis features of TCD, specifically the Coverage Matrix in this example:

80.00% after first 110 test cases

304 of the 380 possible 2-way interactions ?

- Use your mouse / touchpad scrolling to zoom, click and drag to pan when zoomed
- Use the left & right arrow keys to step through coverage
- Press the 'c' key to cycle through available color schemes



Based on your system knowledge, project scope, and risk tolerance, there may be a better stopping point than 100% pairwise coverage. The visualization above will help you find it as, from the tooltip, you will know exactly which interactions are “sacrificed”.



If you are generally happy with the 80% state but there are a few gaps you want to close, Forced Interactions could be used to achieve that efficiently (if moving the slider does not).

**Impact – reduces the count by 42 at 80%.**

Please note – when you export or sync test sets, 100% of your tests will be migrated, so you would need to remove the extra scenarios after the export /sync operation.

## Strategy 2 – Adjust the coverage dial setting away from 2-way

Commonly, parameters in the model do not all have the same level of significance. Therefore, risk-based testing is applied. In Test Case Designer, that feature is called “mixed-strength testing”.

Let’s say, in this example, Applicant Age is as relevant, and getting each value once would be sufficient (i.e., its interactions don’t matter). We can reduce the setting for that parameter to 1-way:

100 scenarios and 114 Mixed-strength interactions

Reapply

2-way
Risk State

2-way
Roof Shape

1-way
Applicant Age

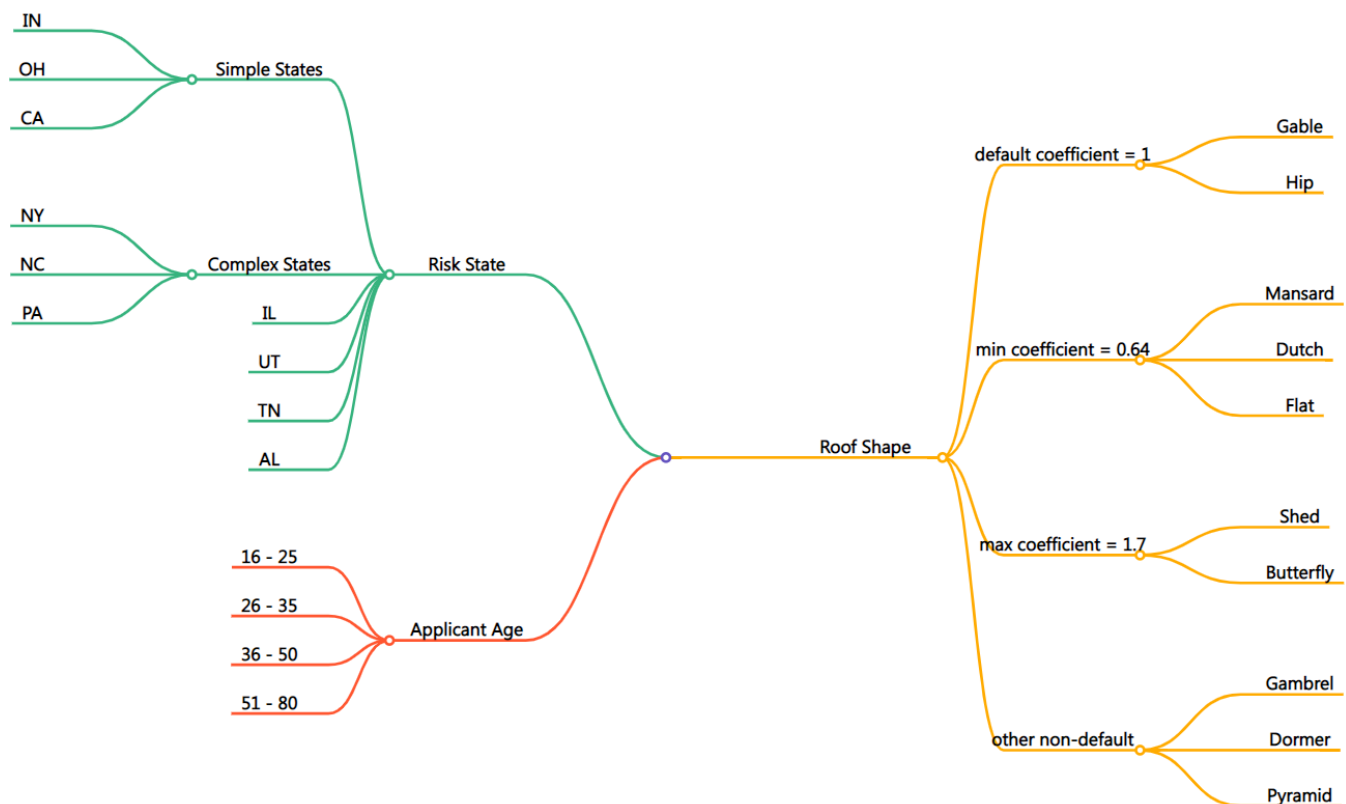
**Impact** – reduces the count by 52 with one largest parameter set to 1-way.

Potential downside – especially early in the project, the exact settings are often “best guesses” based on personal judgments.

## Strategy 3 – Change how you describe your inputs (e.g., with optimizations to Parameters, Values, and/or Value Expansions)

It is also common that not all parameter values are equally important – the system rules can be structured, so that ages within a certain range behave. Similarly, state groups have the same laws, and roof shapes have the same coefficients in the pricing engine.

The TCD features to account for that are the variations of the “Equivalence Classes” approach - Ranged Values and Value Expansions:



Keep the system rules in mind when applying this method, as it is impossible to apply constraints to value expansions in TCD.

**i** It is not required to have all values grouped into expansions or to have ranges cover the whole continuum (i.e., it's ok to have a break between 50 and 55 as long as the ranged values don't overlap).

**Impact** – reduces the count by 128 with the restructure of all parameters

Potential downside – excessively aggressive grouping may lead to some value expansions being completely removed from scenarios (i.e, if there are 10 expansions but 8 scenarios with the value, the last 2 expansions would never be used).

# Strategy 4 – Add "artificial" constraints

The general logic of this method is invalidating interactions that can happen, but we are not interested in them.

For this example, let's say fewer than 1% of our customers a) have Gable roof in IN; b) are 16 y.o. in CA, so we exclude those pairs:

never	Risk State = IN	Risk State (10)	✗ 1 IN	OH	✗ 1 CA
never	Roof Shape = Gable				
never	Risk State = CA	Roof Shape (10)	✗ 1 Gable	Hip	Mansard
never	Applicant Age = 16				
		Applicant Age (14)	✗ 1 16	18	20

Impact – reduces the count by 5 with 2 artificial constraints.

Potential downside – 1) keeping track of which constraints are fake vs. real; 2) implementing enough fake constraints to move the needle in scenario count.

## Conclusion

The methods described above are not mutually exclusive. At the extreme end of the spectrum, you can use all 4 together and settle on 80% of mixed-strength after the parameter restructure and 2 artificial constraints (which results in just 18 scenarios – a count reduction of 134 compared to the benchmark).

The reverse is true – you can boost the thoroughness by increasing the algorithm settings, “unpacking” value expansion into standalone values, etc.

As you can see, the “test case count” metric is very volatile regarding model-based testing, so you don't need to settle for the excessive (or insufficient) count with the default settings. Instead, re-evaluate the model elements and identify which ones need to be tweaked – the impact on the count would often be disproportionately large.

Finding the initial quality/quantity balance often requires at least a few subjective assumptions based on the SME knowledge, so it is important to establish the feedback loop for the longer term and track the execution results, specifically the scenario-to-defect ratio. If it remains consistently in double-digits, it may be a sign to try the methods described in this article.