

How Are TCD Tests Objectively Superior?

One of the most common concerns we hear from clients is:

“Our current tests are good enough. Why invest the time and resources in adopting the Test Case Designer methodology if it doesn’t move the needle?”

That may be true, but we don’t have to guess. In this document, we describe the process of evaluating the existing tests, directly compare them with the tests generated by TCD, and make data-driven conclusions about what is best for the software testing efficiency in your organization.

The example below uses a simple banking application. Still, this process can also be followed with any existing set of tests as long as it is converted to the parameterized data table. The order of creating the optimized Test Case Designer model or the one for analyzing the existing suite technically doesn't matter (this article goes through the manual side first). **It is crucial, though, that the models are exactly the same regarding Parameters/ Values and value expansions/ Constraints.**

 The process description assumes intermediate knowledge of Test Case Designer features.

- [Prerequisites](#)
- [Process](#)
- [Comparison & Conclusions](#)


Prerequisites

You will need to reorganize the existing tests in the TCD import format. It involves Parameters as columns and Values as rows:

Name	Description	Expected Outcome	Type of Income	Credit Rating	Customer Status	Term of Loan	Loan Amount	Type of Property	Loan to Value Ratio	Location of Property
Rapid Approval	When all the following factors are present, then the application gets instantly approved	High	High	Employee	20 yrs	10,000 - 99,999	Apartment	80%	Small City	
Historically Problem Area	When the application relates to a condo in a rural area, then route it to the Sr. Underwriter or her assistant.	Medium	Medium	Regular	20 yrs	100,000 - 199,999	Condo	80%	Rural Area	
PMI	When more than 80% of a home's value is borrowed, then request private mortgage insurance	Medium	Medium	Regular	30 yrs	200,000 - 500,000	Apartment	90%	Small City	
Additional Scrutiny	When the customer has the low credit rating, then request all additional documents	Medium	Low	Regular	20 yrs	100,000 - 199,999	Apartment	80%	Small City	
Rapid Denial	When all 3 of these things are present, then the application gets instantly denied	Low	Low	Regular	20 yrs	200,000 - 500,000	Apartment	80%	Small City	
Elite Mortgages	When a VIP client applies for a house in big city, then it should be routed to the Elite Mortgage manager.	Medium	Medium	VIP	20 yrs	100,000 - 199,999	House	80%	Big City	
Reduced Rate	When the following 4 conditions are met, then provide 0.45% discount to the annual rate	Medium	Medium	Regular	20 yrs	100,000 - 199,999	Apartment	70%	Small City	
First Line of Mortgage	When a regular client opens the first line of mortgage at 80% LTV ratio, then provide optimal loan amount FAQ link	Medium	Medium	Regular	20 yrs	100,000 - 199,999	Apartment	80%	Small City	

Most requirements and existing tests specify **only** the parameters necessary to trigger the outcome (i.e., black font above). While it can be beneficial for precise impact identification, it lacks the systematic approach to selecting the values for other parameters and leaves room for the redundancy of the manual process at the tester’s discretion. Traditional approaches leave plenty of room for the challenges below:

- Direct duplicates (inconsistent formatting; spelling errors);
- “Hidden”/Contextual duplicates (meaningful typos; same instructions written by different people with varied styles);
- Tests specifying some values, leaving others as default (when several scenario combinations could be tested in the single execution run).

 There is a difference between “select these 3 values, and everything else should be default for this rule” and “select these 3 values, and everything else can be anything because it doesn’t matter for this rule”. The second interpretation is much more common in our experience.

To generate the most precise comparison, the actual values from execution logs need to be placed in all the blanks in requirements (i.e., red font in the picture above). If that is impossible, the default value for each parameter is assumed to be used.

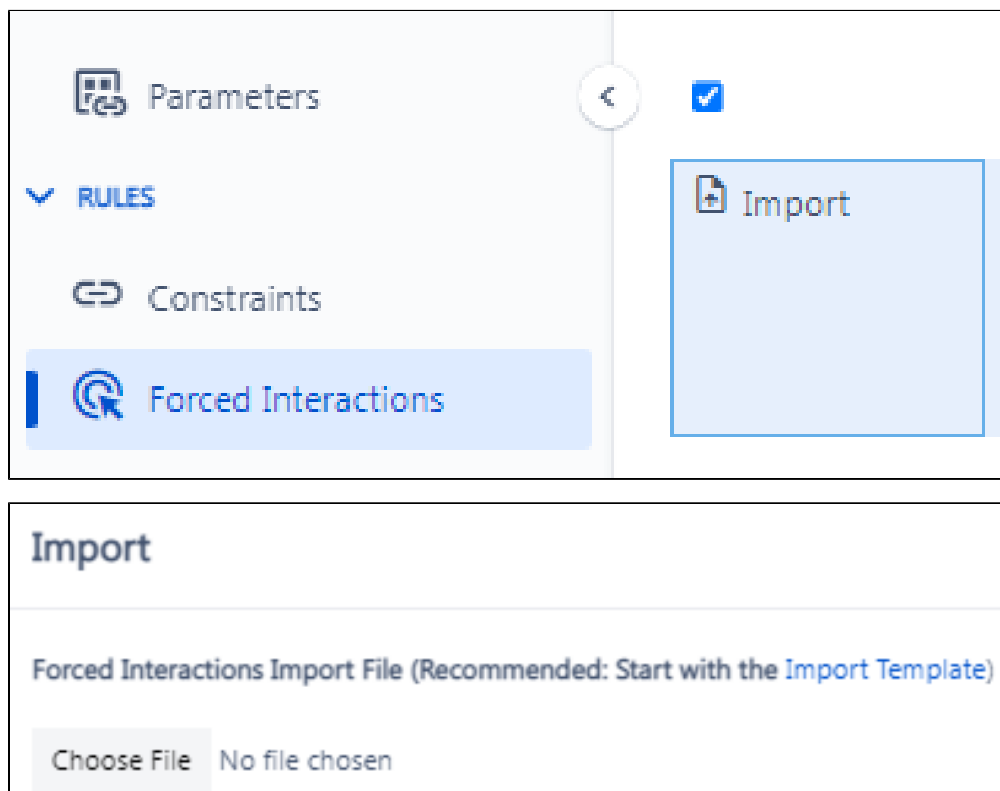
For this example, we use 8 artificial existing tests which didn't specify all the values in their documentation.

Process

Next, we proceed with generating the comparison. First, we create the model to put the reformatted dataset above onto the Forced Interactions tab. There are 2 options:

- Manually create a model with all the parameters & values, then input each of the existing tests inside the tool on the Forced Interactions tab.
- Create an empty model with 2 dummy parameters, navigate to Forced Interactions, and use the cloud icon on the left for the Import dialog. It contains the template you should copy-paste the reformatted existing tests to.

Note: when working with large existing suites, making the updates in Excel and importing the file into Test Case Designer is generally faster, but let us know if you run into any issues.



Verify the "Forced Interactions" screen looks like the following, with each scenario specifying all parameters/values necessary for the execution:

Name	Description	Forced Parameter Values	Expected Outcome
Rapid Approval		Type of Income = High Credit Rating = High Customer Status = Employee Term of Loan = 20 yrs	When all the following factors are present, then the application gets instantly approved
Historically Problem Area		Type of Income = Medium Credit Rating = Medium Customer Status = Regular Term of Loan = 20 yrs	When the application relates to a condo in a rural area, then rout it to the Sr. Underwriter or her assistant.
PMI		Type of Income = Medium Credit Rating = Medium Customer Status = Regular Term of Loan = 30 yrs	When more than 80% of a home's value is borrowed, then request private mortgage insurance
Additional Scrutiny		Type of Income = Medium Credit Rating = Low Customer Status = Regular Term of Loan = 20 yrs	When the customer has the low credit rating, then request all additional documents
Rapid Denial		Type of Income = Low Credit Rating = Low Customer Status = Regular Term of Loan = 20 yrs	When all 3 of these things are present, then the application gets instantly denied
Elite Mortgages		Type of Income = Medium Credit Rating = Medium Customer Status = VIP Term of Loan = 20 yrs	When a VIP client applies for a house in big city, then it should be routed to the Elite Mortgage manager.
Reduced Rate		Type of Income = Medium Credit Rating = Medium Customer Status = Regular Term of Loan = 20 yrs	When the following 4 conditions are met, then provide 0.45% discount to the annual rate
First Line of Mortgage		Type of Income = Medium Credit Rating = Medium Customer Status = Regular Term of Loan = 20 yrs	When a regular client opens the first line of mortgage at 80% LTV ratio, then provide optimal loan amount FAQ link

Next, click "Scenarios" in the left navigation pane.


Search...

19 scenarios and 252

2-way interactions

▼

#	Type of Income	Credit Rating	Customer Status	Term of Loan	Loan Amount
1	High	High	Employee	20 yrs	10,000
2	Medium	Medium	Regular	20 yrs	160,592
3	Medium	Medium	Regular	30 yrs	200,000
4	Medium	Low	Regular	20 yrs	100,000

 The process is the same for any N-way strength; this example covers 2-way for simplicity and the easy availability of the coverage matrix.

This is how your existing test suite looks when "generated" by TCD. However, the algorithm believes you need 19 test cases (not the 8 we imported in this example) to explore the potential system weaknesses thoroughly. Why?

Before diving into that, copy the model you just created, remove all forced interactions in that second version, and generate the scenarios there.

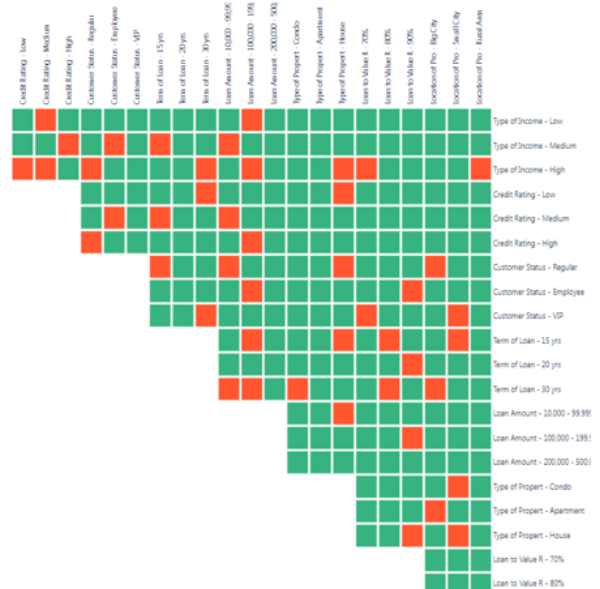
Then the answer to the central question of this guide is on the Analysis screen.

Comparison & Conclusions

8 Hand-Selected Tests



12 Tests (8 manual + 4 TCD)



Remember the dangers of manual selection without a systematic approach? The "good enough" existing suite only covers 48% of 2-way interactions in the system, leaving a significant number of potentially-defect-causing gaps in coverage.

Granted, the more experienced testing organizations focusing on variations and with some knowledge of combinatorial methodologies will do better than this. Yet, it is rare that manual selection can consistently achieve the coverage levels in the second picture.

Thus, this portion of the comparison tells us that the existing thoroughness is insufficient, and 4 more TCD-generated tests would be needed to get to 81% 2-way interactions, a safe benchmark proven by research studies. You can clearly see which pairs are still missing and make concrete execution decisions based on the business risks & constraints (i.e., execute all 19 tests to reach 100%).

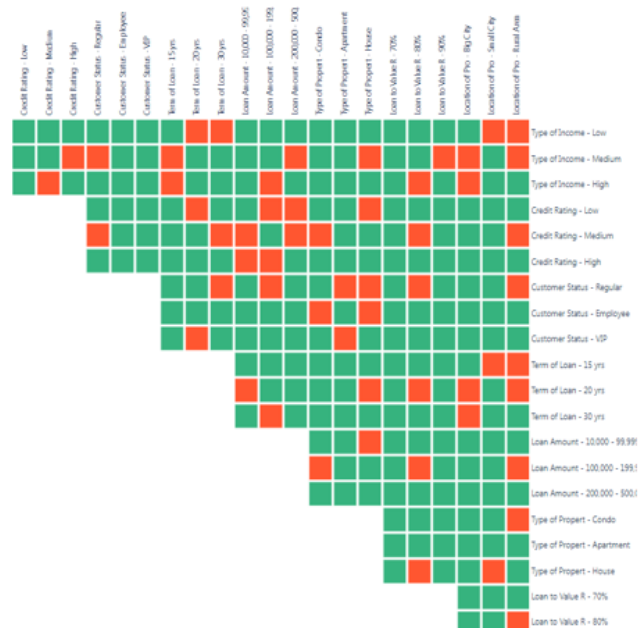
However, that is **not** the key conclusion. These 2 images evaluate the concept of building TCD tests on top of the existing ones just to close the coverage gaps. This approach ignores the potential benefits of completely remodeling the application inside the Test Case Designer. Let's prove the benefits of this alternate approach by looking at the model we copied (with removed forced interactions).

What if you let Test Case Designer select all the non-specified values for the 8 business rules that you had? As you go to the Analysis tab in that copied model, this is what you should notice:

Before (without TCD)



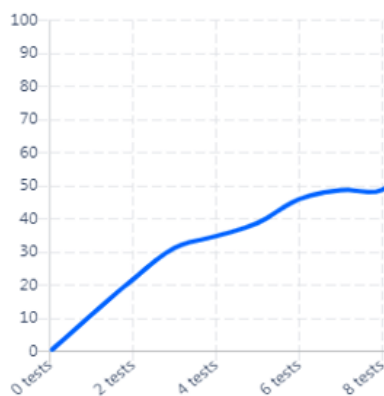
After (with TCD)



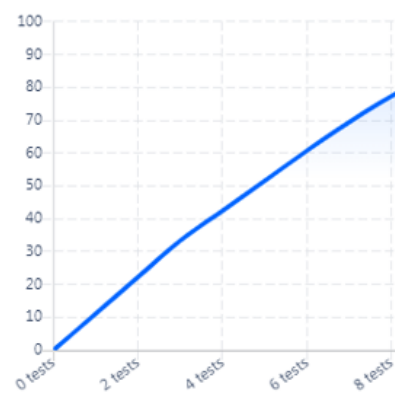
We recommend opening the models in 2 different browser tabs so you can easily go back and forth.

Test Case Designer can scientifically detect the optimal way to select values for each test scenario and generate 26% more interaction coverage with the same number of tests. Consequently, you hit the diminishing returns on coverage a lot sooner, and your total suite size is smaller (17 in this case).

Before (without TCD)



After (with TCD)



This is the process to prove the objectively-superior nature of the TCD-selected tests. Did your results come out drastically different than the above ones? Please feel free to contact us and share your experience or ask for advice on putting together this comparison yourself.