

Decision Table Testing: Learn with Example



تداول على الإنترنت
ابدأ مع \$30 بونص تداول*

اقرأ المزيد

*تطبيق الشروط والأحكام. قد يؤدي تداول الفوركس وعقود الفروقات إلى مخاطر عالية ويمكن أن يتسبب في خسارة كل رأس المال المستثمر.

USAIN BOLT OFFICIAL SPONSOR | XM

What is Decision Table Testing?

Decision table testing is a testing technique used to test system behavior for different input combinations. This is a systematic approach where the different input combinations and their corresponding system behavior (Output) are captured in a tabular form. That is why it is also called as a **Cause-Effect** table where Cause and effects are captured for better test coverage.

A Decision Table is a tabular representation of inputs versus rules/cases/test conditions. Let's learn with an example.

Example 1: Decision Base Table for Login Screen

Let's create a decision table for a login screen.



Email

Password

Log in

[.\(\(cdn.guru99.com/images/1/120817_0759_DecisionTab1.png\)](http://cdn.guru99.com/images/1/120817_0759_DecisionTab1.png)

The condition is simple if the user provides correct username and password the user will be redirected to the homepage. If any of the input is wrong, an error message will be displayed.

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Username (T/F)	F	T	F	T
Password (T/F)	F	F	T	T
Output (E/H)	E	E	E	H

Legend:

- T – Correct username/password
- F – Wrong username/password
- E – Error message is displayed
- H – Home screen is displayed

Interpretation:

- Case 1 – Username and password both were wrong. The user is shown an error message.
- Case 2 – Username was correct, but the password was wrong. The user is shown an error message.

- Case 3 – Username was wrong, but the password was correct. The user is shown an error message.
- Case 4 – Username and password both were correct, and the user navigated to homepage

While converting this to test case, we can create 2 scenarios ,

- Enter correct username and correct password and click on login, and the expected result will be the user should be navigated to homepage

And one from the below scenario

- Enter wrong username and wrong password and click on login, and the expected result will be the user should get an error message
- Enter correct username and wrong password and click on login, and the expected result will be the user should get an error message
- Enter wrong username and correct password and click on login, and the expected result will be the user should get an error message

As they essentially test the same rule.

Example 2: Decision Base Table for Upload Screen

Now consider a dialogue box which will ask the user to upload photo with certain conditions like –

1. You can upload only '.jpg' format image
2. file size less than 32kb
3. resolution 137*177.

If any of the conditions fails the system will throw corresponding error message stating the issue and if all conditions are met photo will be updated successfully

*upload .jpg file with size not more than 32kb and resolution 137*177

(http://cdn.guru99.com/images/1/120817_0759_DecisionTab2.png)

[i](#) [x](#)

Let's create the decision table for this case.

Conditions	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Format	.jpg	.jpg	.jpg	.jpg	Not .jpg	Not .jpg	Not .jpg	Not .jpg
Size	Less than 32kb	Less than 32kb	>= 32kb	>= 32kb	Less than 32kb	Less than 32kb	>= 32kb	>= 32kb
resolution	137*177	Not 137*177	137*177	Not 137*177	137*177	Not 137*177	137*177	Not 137*177
Output	Photo uploaded	Error message resolution mismatch	Error message size mismatch	Error message size and resolution mismatch	Error message for format mismatch	Error message format and resolution mismatch	Error message for format and size mismatch	Error message for format, size, and resolution mismatch

For this condition, we can create 8 different test cases and ensure complete coverage based on the above table.

1. Upload a photo with format '.jpg', size less than 32kb and resolution 137*177 and click on upload. Expected result is Photo should upload successfully
2. Upload a photo with format '.jpg', size less than 32kb and resolution not 137*177 and click on upload. Expected result is Error message resolution mismatch should be displayed

3. Upload a photo with format '.jpg', size more than 32kb and resolution 137*177 and click on upload. Expected result is Error message size mismatch should be displayed
4. Upload a photo with format '.jpg', size less than 32kb and resolution not 137*177 and click on upload. Expected result is Error message size and resolution mismatch should be displayed
5. Upload a photo with format other than '.jpg', size less than 32kb and resolution 137*177 and click on upload. Expected result is Error message for format mismatch should be displayed
6. Upload a photo with format other than '.jpg', size less than 32kb and resolution not 137*177 and click on upload. Expected result is Error message format and resolution mismatch should be displayed
7. Upload a photo with format other than '.jpg', size more than 32kb and resolution 137*177 and click on upload. Expected result is Error message for format and size mismatch should be displayed
8. Upload a photo with format other than '.jpg', size more than 32kb and resolution not 137*177 and click on upload. Expected result is Error message for format, size and resolution mismatch should be displayed

Why is Decision Table Testing is important?

This testing technique becomes important when it is required to test different combination. It also helps in better test coverage for complex business logic.

Boundary value and equivalent partition are other similar techniques used to ensure better coverage. They are used if the system shows the **same** behavior for a large set of inputs. However, in a system where for each set of input values the system behavior is **different**, boundary value and equivalent partitioning technique are not effective in ensuring good test coverage.

In this case, decision table testing is a good option. This technique can make sure of good coverage, and the representation is simple so that it is easy to interpret and use.

This table can be used as the reference for the requirement and for the functionality development since it is easy to understand and cover all the combinations.

The **significance of this technique becomes immediately clear as the number of inputs increases. Number of possible Combinations is given by 2^n , where n is the number of Inputs. For $n = 10$** , which is very common in the web based testing, having big input forms, the number of **combinations will be 1024**. Obviously, you cannot test all but **you will choose a rich sub-set of the possible combinations using decision based testing technique**

Advantages of decision table testing

When the system behavior is different for different input and not same for a range of inputs, both equivalent partitioning, and boundary value analysis won't help, but decision table can be used.

The representation is simple so that it can be easily interpreted and is used for development and business as well.

This table will help to make effective combinations and can ensure a better coverage for testing

Any complex business conditions can be easily turned into decision tables

In a case we are going for 100% coverage typically when the input combinations are low, this technique can ensure the coverage.

Disadvantages of decision table testing

The main disadvantage is that when the number of input increases the table will become more complex

Please be patient. The Video will load in some time. If you still face issue viewing video click [here \(/faq.html#1\)](/faq.html#1)

◀ Prev (</equivalence-partitioning-boundary-value-analysis.html>) [Report a Bug](#) Next ▶ (</software-testing-techniques-2.html>)

YOU MIGHT LIKE:

SOFTWARE TESTING

(</system-integration-testing.html>)



(</system-integration-testing.html>)

**System INTEGRATION Testing (SIT):
Complete Tutorial**

SOFTWARE TESTING

(</smoke-testing.html>) (</smoke-testing.html>)



Smoke Testing Learn with Examples
(</smoke-testing.html>)

SOFTWARE TESTING

(</loop-testing.html>) (</loop-testing.html>)



**Loop Testing Tutorial - Types, Strategy,
Limitation & Examples**

[\(/system-integration-testing.html\)](/system-integration-testing.html)

SOFTWARE TESTING



[\(/waterfall-vs-agile.html\)](/waterfall-vs-agile.html)

[\(/waterfall-vs-agile.html\)](/waterfall-vs-agile.html)

Waterfall Vs. Agile: Must Know Differences

[\(/waterfall-vs-agile.html\)](/waterfall-vs-agile.html)

SOFTWARE TESTING



[\(/volume-testing.html\)](/volume-testing.html)

[\(/volume-testing.html\)](/volume-testing.html)

Volume Testing Tutorial: Learn with Examples

[\(/volume-testing.html\)](/volume-testing.html)

[\(/loop-testing.html\)](/loop-testing.html)

SOFTWARE TESTING



[\(/dynamic-testing.html\)](/dynamic-testing.html)

[\(/dynamic-testing.html\)](/dynamic-testing.html)

Dynamic Testing Tutorial: Types, Techniques & Process

[\(/dynamic-testing.html\)](/dynamic-testing.html)

Testing Tutorials

[22\) BVA & EP \(/equivalence-partitioning-boundary-value-analysis.html\)](/equivalence-partitioning-boundary-value-analysis.html)

[23\) Decision Table Testing \(/software-testing-techniques-1.html\)](/software-testing-techniques-1.html)

[24\) State Transition \(/software-testing-techniques-2.html\)](/software-testing-techniques-2.html)

[25\) Use Case Testing \(/software-testing-techniques-3.html\)](/software-testing-techniques-3.html)

[26\) Estimation Techniques \(/testing-estimation.html\)](/testing-estimation.html)

[27\) Test Plan Document \(/test-plan.html\)](/test-plan.html)

[28\) Test Plan Template \(/test-plan-for-project.html\)](/test-plan-for-project.html)

[29\) Test Environment \(/test-environment-software-testing.html\)](/test-environment-software-testing.html)

[30\) What is Bug \(/the-unconventional-guide-to-defect-management.html\)](/the-unconventional-guide-to-defect-management.html)

[31\) Bug Life Cycle \(/defect-life-cycle.html\)](/defect-life-cycle.html)

[32\) Testing Tools \(/testing-tools.html\)](/testing-tools.html)

[33\) Agile Methodology \(/agile-scrum-extreme-testing.html\)](/agile-scrum-extreme-testing.html)

- [34\) Agile Testing_\(/agile-testing-a-beginner-s-guide.html\)](#)
- [35\) Scrum Testing_\(/scrum-testing-beginner-guide.html\)](#)
- [36\) Agile Methodology_\(/a-discussion-on-automation-testing-with-respect-to-agile-methodology.html\)](#)
- [37\) Scaled Agile Framework\(SAFE\)_\(/scaled-agile-framework.html\)](#)
- [38\) Waterfall Vs. Agile_\(/waterfall-vs-agile.html\)](#)
- [39\) Bug/Defect Triage_\(/bug-defect-triage.html\)](#)
- [40\) Agile Vs Scrum_\(/agile-vs-scrum.html\)](#)
- [41\) Scrum Master Training_\(/scrum-master-training.html\)](#)
- [42\) Scrum Vs. Kanban_\(/scrum-vs-kanban.html\)](#)
- [43\) Agile Vs. DevOps_\(/agile-vs-devops.html\)](#)
- [44\) Agile Vs Kanban_\(/agile-vs-kanban.html\)](#)
- [45\) Web Testing_\(/web-application-testing.html\)](#)
- [46\) Web Testing Checklist_\(/complete-web-application-testing-checklist.html\)](#)
- [47\) Banking Application Testing_\(/banking-application-testing.html\)](#)
- [48\) Payment Gateway Testing_\(/payment-gateway-testing-tutorial-with-sample-test-cases.html\)](#)
- [49\) Mainframe Testing_\(/mainframe-testing.html\)](#)
- [50\) Testing for Retail POS System_\(/testing-for-retail-pos-point-of-sale-system.html\)](#)

About

[About US \(/about-us.html\)](/about-us.html)

[Advertise with Us \(/advertise-us.html\)](/advertise-us.html)

[Write For Us \(/become-an-instructor.html\)](/become-an-instructor.html)

[Contact US \(/contact-us.html\)](/contact-us.html)

Career Suggestion

[SAP Career Suggestion Tool \(/best-sap-module.html\)](/best-sap-module.html)

[Software Testing as a Career \(/software-testing-career-complete-guide.html\)](/software-testing-career-complete-guide.html)

[Top Tools List \(/testing-development-tools.html\)](/testing-development-tools.html)

[Certificates \(/certificate-it-professional.html\)](/certificate-it-professional.html)

Interesting

[Books to Read! \(/books.html\)](/books.html)

[Suggest a Tutorial](#)

[Blog \(/blog/\)](/blog/)

[Quiz \(/tests.html\)](/tests.html)

Execute online

[Execute Java Online \(/try-java-editor.html\)](/try-java-editor.html)

[Execute Javascript \(/execute-javascript-online.html\)](/execute-javascript-online.html)

[Execute HTML \(/execute-html-online.html\)](/execute-html-online.html)

[Execute Python \(/execute-python-online.html\)](/execute-python-online.html)

© Copyright - Guru99 2018

[Privacy Policy \(/privacy-policy.html\)](/privacy-policy.html)