

KPI Analysis

Testing KPI should not be reduced to only one. A fundamental strength of white-box testing techniques and black-box techniques is that they complement each other and they increase coverage measures. Performing only one technique does not provide an objective measure of actual code coverage.

Branch Coverage:

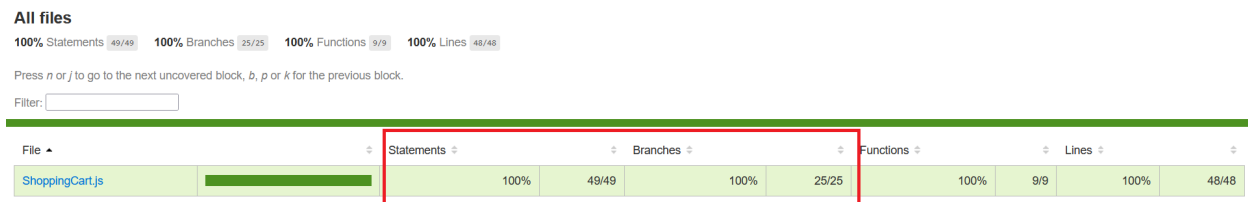
This measure helps to ensure that your code behaves as expected under different conditions. It helps to detect defects or vulnerabilities with unexpected inputs. Branch coverage does not guarantee that your code is defect-free. It does not tell you if the code meets the users needs. Therefore, it should be complemented with other types of measures.

Branch coverage is measured as the number of branches exercised by the test divided by the total number of branches in the code. This KPI is expressed as a percentage. When 100% is covered all branches on the code were exercised.

$$\% = (\text{number of branches executed} \div \text{number of branches in the code}) \times 100$$

In our code we achieve 100% of branch coverage implementing unit testing.

$$(25 \div 25) \times 100 = 100\%$$



This type of coverage includes **Statement coverage**, this means that achieving 100% branch coverage it also achieves 100% statement coverage. If we only execute 100% of statement coverage it does not mean that 100% branch coverage was achieved.

Passed Test Case coverage:

This measure calculates the percentage of success in the most recent software testing run. This helps to understand the status of the current test with the previous test. For example, if the percentage of success decreases from the last test run, without introducing new test cases, this means new defects were introduced on the new features or fixes.

Passed test case coverage is measured as the number of test cases that executed successfully divided by the total number of test cases.

$$\% = (\text{number of passed test cases} \div \text{total number of test cases}) \times 100$$

In our end to end test we achieve 100% of passed test coverage. Given that we only have a test case and it passed in all the executions.

$$(1 \div 1) \times 100 = 100\%$$

```

Tests:      1
Passing:    1
Failing:    0
Pending:    0
Skipped:    0
Screenshots: 0
Video:      false
Duration:    9 seconds
Spec Ran:    shoppingCart.cy.js

```

Compatibility coverage:

This measure calculates the percentage of success to run in different browsers, databases, hardware, operating system, mobile devices, etc. It helps to detect compatibility related problems and is oriented to non functional tests.

Compatibility coverage is measured as the number of tests that executed successfully divided by the total number of tests.

$$\% = (\text{number of passed tests} \div \text{total number of tests}) \times 100$$

In this test run we only have 1 test case that was tested on 4 different browsers (chrome, firefox, Edge, Electron). Therefore, the total number of tests was 4. The number of passed tests were 3, failing on firefox, having a 75% of browser compatibility coverage.

$$(3 \div 4) \times 100 = 75\%$$