### What is Code Coverage?

Code coverage is a measure, which describes the degree of which the source code of the program has been tested.

It is one form of white box testing which finds the areas of the program not exercised by a set of test cases.

### Why use code coverage?

* It helps you to measure the efficiency of test implementation
* It offers a quantitative measurement.
* It defines the degree to which the source code has been tested.

### Code Coverage Methods

* Statement Coverage
* Decision Coverage
* Branch Coverage

**Statement Coverage** is a white box testing technique in which all the executable statements in the source code are executed at least once. It is used for calculation of the number of statements in source code, which have been executed. The main purpose of Statement Coverage is to cover all the possible paths, lines and statements in source code.

What is covered by Statement Coverage?

* Unused Statements
* Dead Code
* Unused Branches
* Missing Statements

**Decision Coverage** is a white box testing technique which reports the true or false outcomes of each boolean expression of the source code. The goal of decision coverage testing is to cover and validate all the accessible source code by checking and ensuring that each branch of every possible decision point is executed at least once.

**Branch Coverage** is a white box testing method in which every outcome from a code module(statement or loop) is tested. The purpose of branch coverage is to ensure that each decision condition from every branch is executed at least once. It helps to measure fractions of independent code segments and to find out sections having no branches.

For example, if the outcomes are binary, you need to test both True and False outcomes.

***Advantages of Branch coverage:***

* Allows you to validate-all the branches in the code
* Helps you to ensure that no branched lead to any abnormality of the program's operation
* Branch coverage method removes issues which happen because of statement coverage testing
* Allows you to find those areas which are not tested by other testing methods
* It allows you to find a quantitative measure of code coverage
* Branch coverage ignores branches inside the Boolean expressions

### Advantages of Using Code Coverage

* Helpful to evaluate a quantitative measure of code coverage
* It allows you to create extra test cases to increase coverage
* It allows you to find the areas of a program which is not exercised by a set of test cases

### .Net Code Coverage Tools

Visual Studio code coverage - Available only in enterprise edition

* It reports coverage percentages at various granularities (e.g., assembly, class, etc.).
* You can select all of your tests or subsets of them.
* You can have it paint your IDE, meaning it lets you actually visualize the coverage as you look at your code.
* It comes from Microsoft, so you can expect plenty of maintenance and support if you invest in its usage.

#### NCover

* Ncover is third party tool for visual studio. It is not the free tool.
* NCover provides extremely detailed information, not only supporting measurements of coverage but also integrating it for the whole team. It tracks trends in coverage and provides detailed reporting.
* NCover is a comprehensive tool for this purpose. Some features include
* Detailed and centralized data about coverage.
* Extensive documentation and user support.
* 32- and 64-bit support, plus memory consumption optimization.
* IDE painting.
* Reports can be obtained in the following formats: XML, HTML, JSON

DotCover - Shipped along with Resharper (static code analysis tool)

* JetBrains dotCover comes with ReSharper Ultimate, which costs $399 per year.
* Detailed reporting.
* Both Visual Studio and CI integration of coverage measurements.
* Navigate to covering tests.
* A cool “hotspots” view that calls out risky methods.
* IDE painting.

OpenCover - Open source and free of cost

OpenCover supports .NET 2 and above (only on Windows) and works with both 32-bit and 64-bit processors, and you can get it via NuGet.

Features include

* 32- and 64-bit support.
* Both branch and statement coverage support (two subtly different ways to measure coverage).
* Generation of an XML-based report that, combined with ReportGenerator, produces a nice HTML-based report on coverage.
* Free and open source, so you can adjust to suit your needs.

### References:

<https://www.guru99.com/code-coverage.html>

<https://chandradev819.wordpress.com/category/tools/>

<https://visualstudio.microsoft.com/vs/compare/>

<https://blog.ndepend.com/guide-code-coverage-tools/>