Code coverage
OpenCover
Motivation for testing

- People make mistakes..
  - leading to defects and errors in products

- Tests reduce:
  - bugs in existing / added code
  - time (money) needed to fix bugs
  - fear of making changes

- Tests also help with:
  - documentation
  - constraining features
Testing shows the presence, not the absence of bugs.
Code coverage

- Measures lines covered by test cases
- Various subtypes
  - statement coverage
  - method coverage
  - branch coverage
  - ...

- Straightforward process
  - add tests and check if code coverage improved
What code coverage tells you

- Finds code that has NOT been tested at all
- What lines have been executed by certain tests
Disadvantages

- Certain level of code coverage does not necessarily mean the test suite is effective.

- In other words: high code coverage is not a sufficient approach when measuring effectiveness of a test suite.
Open source code coverage tool
Supporting .NET 2 and above (and Silverlight)
Command-line interface

GitHub link (also available as Nuget)
Windows only
Command-line arguments

- Mandatory:
  - \texttt{target:<path to the target>} Path to the target application or service to start

- Optional:
  - \texttt{targetargs:"arg1 \ldots argN"} Additional arguments to pass on application start
  - \texttt{output:<output file>} Path for coverage results
  - \texttt{register[::user]} Registering assemblies
Generating test coverage

@echo off
SET openCover=OpenCover.4.6.519
"%%~dp0\packages\%openCover%\tools\OpenCover.Console.exe" ^
-register:user ^
-target:"%VS140COMNTOOLS%\..\IDE\MSTest.exe" ^
-targetargs:"/testcontainer:"%%~dp0\Tests\bin\Debug\Tests.dll"" ^
-output:"%%~dp0\GeneratedReports\testReport.xml"
Example
Transforming XML output from OpenCover

GitHub link (also available as Nuget)
Command line arguments

- **Mandatory**
  - `-reports:<report>[;<report>]` coverage reports to process separated by semicolons
  - `-targetdir:<target directory>` determines where to save generated report

- **Optional**
  - `-historydir:<history directory>` storing persistent coverage information
  - `-tag:<tag>` additional information like build version
Generating reports

@echo off
SET reportGenerator=ReportGenerator.3.1.2

"%~dp0\packages\%reportGenerator%\tools\ReportGenerator.exe" ^
-reports:"%~dp0\GeneratedReports\testReport.xml" ^
-targetdir:"%~dp0\GeneratedReports\CodeCoverage" ^
-historydir:"%~dp0\GeneratedReports\History"

start “Report" "%~dp0\GeneratedReports\CodeCoverage\index.htm"
Example