



THE LEADING .NET CODE QUALITY PLATFORM

The Actual Cost Benefits of NCover Code Coverage Software

September 2009



Code Coverage Background

WHAT IS CODE COVERAGE

Code coverage analysis is a measurement that is taken on running program code, reporting how much of that code was executed. Specifically, code coverage usually tells you how many times each line of code was executed. There are various forms of coverage metrics, such as function coverage, sequence point coverage, statement coverage and branch coverage.

Code coverage is most valuable as a feedback mechanism for test-driven or agile development methodologies.

IMPORTANCE OF CODE QUALITY

Code coverage is most valuable as a feedback mechanism for test-driven or agile development methodologies. Both of these methods rely on a developmental feedback loop that promotes the addition of features while maintaining a predictable quality level. Code quality is one of the most important concerns of any software development organization. Rapid development and technical wizardry will do you no good if customers can not rely on the quality of your product.

UNIT TEST SUITES

The way most modern software development teams maintain the quality of their software is by building a suite of unit tests that automate the verification of code correctness. These unit test suites are designed to run through often thousands of simulated program functions, verifying that the code is producing the expected output. While creating a number of unit tests is laudable, just creating a suite of tests alone is not enough. After some point, adding more tests does not necessarily mean that the code has better quality. You may just be testing the same code over and over again, while missing an important piece of code entirely. Any bugs in code that is not tested will not be detected by your unit test suite and thus could creep into production.

Code coverage reports allow developers to quickly find code that is not executed by the test suite

COVERAGE ANALYSIS

Coverage analysis closes that feedback loop by reporting on the comprehensiveness of your unit tests. Code coverage reports allow developers to quickly find code that is not executed by the test suite.



INTEGRATING NCOVER

Harnessing this deep understanding of coverage analysis, NCover empowers developers to predictably deploy quality software using the latest in multiprotocol coverage analytics and visual trending analysis. Studies indicate that end users of software created using the NCover suite of tools appreciate the added performance and piece-of-mind from reliable solutions.

NCover empowers developers to predictably deploy quality software using the latest in multiprotocol coverage analytics and visual trending analysis.

Code coverage reports allow developers to quickly find code that is not executed by the test suite

The end result of integrating NCover into the development environment is seen quite clearly in higher quality code, quicker Quality Assurance problem highlighting, and faster product development timelines. Numerous research has pinpointed each of these areas as seeing significant benefit from code coverage instrumentation. Discussed less frequently but perhaps of even greater value is the bottom line benefit of code coverage.

NCover for .NET Development Provides Significant Bottom Line Benefit

Annual Savings:

\$5500

First-year return on investment:

836%*

* The information presented in this report is based on dual research by NCover, Inc. and information obtained through a comprehensive study conducted by an independent consultant for IBM. More than forty companies in different business verticals contributed to the findings. Most were employed by large organizations with over 500 employees and revenues over \$100 million annually.



BENEFITS OF NCOVER

The table on the previous page demonstrates that **NCover** not only saves a company valuable time and resources but also provides a quantifiable return on invested purchase. Let's explore this further.

IMPROVING ERROR DETECTION

Code coverage products like **NCover** improve error detection in the product development stage by proactively pinpointing areas of code that have not been sufficiently tested or where overly complex code could indicate potential bugs or performance issues. This early detection of potential buggy code significantly reduces the risk of depleting your expectant bottom line through expensive post-shipping bug fixes.

COST OF DEFECTS

Research has shown that the time and costs associated with fixing defects in software grow exponentially as the product moves beyond the development stage to the shipping life cycle. The generally accepted multiplier for

this cost differential is a factor of 10 or 1000%. That means that a \$1500 pre-shipping bug fix becomes a \$15,000 disaster when discovered once the product is already in user's hands.

BUG DETECTION

On average, developers implementing **NCover** tools into their product development life-cycle will catch an additional ten bugs per developer per year in contrast to developers not using the product or their previous workflow before using the **NCover** approach.

Savings with NCover

10 bugs (discovered annually)

X **\$550** (cost to fix after shipping)

\$5500



SAVINGS WITH NCOVER

The associated costs of fixing an average bug after the software had shipped to the customer was estimated to be \$550. This compares to \$55 to fix the same problem using NCover in the development life-cycle. By simply catching problems earlier, **NCover** saves a company per developer \$5,500 per year.

SYNERGISTIC ADVANTAGES

These numbers do not include the synergistic advantages of an improved workflow and testing process refinement of your continuous integration development. Those advantages, while of seemingly

significant value, are much harder to quantify. The raw numbers overwhelmingly indicate that test coverage analysis provides tangible bottom line value and should be considered a primary development tool in creating profitable products. From a proactive risk mitigation position, **NCover** provides a more reliable baseline for anticipated return on investment. By adding **NCover** to your product development life-cycle, you will be better positioned to improve your software reliability, to stabilize your quality assurance efforts, and increase the individual productivity of your developers. Most importantly, you will save money.

NCover Case Study Assumptions

Price of NCover Software (includes support):	\$658
Annual Savings:	\$5500
First-year return on investment:	836%

For more information about how NCover can boost your bottom line, visit our website at www.NCover.com.