

Diffblue Cover Refactor: Improve The Testability Of Java Code

Use AI for Code to increase unit test coverage by automatically updating existing code.

Automatic code improvement

Diffblue Cover Refactor is a code optimization tool that improves unit test coverage and reduces the need for unproductive developer effort. Refactor automatically improves the testability of Java source code to decrease the risk of errors when changes are made.

There's some code for which unit tests simply cannot be written, even by the most experienced Java developer. It's not a matter of skill: the code in question is simply untestable in its current form. Perhaps a method isn't fully observable, nor completely deterministic; maybe it relies on behavior that can't be defined until runtime.

This untestable code is a problem. There's no way to use the power and speed of unit testing to identify regressions, which means risk for your organization.

Cover Refactor's initial release uses AI to suggest and apply code refactorings that improve the observability of Java code and make it more testable. It can even make these changes automatically, without any input from your team, letting you reduce the risk of change and get more value from unit testing. Future updates are planned to expand the capability of this powerful technology.

Increase unit test coverage

The first step to improving testability is understanding how much of your code can be tested now - traditionally no easy task. Cover's core AI for Code engine executes deep analysis of Java source code, then visualizes how testable your code is and where risk lies. This analysis is the basis for Refactor's automatic code improvements.

But Refactor doesn't just make code more testable - it also increases code coverage by default. Unit tests for newly testable code are automatically written the next time Cover is used to update your unit test suite.



COVER REFACTOR SPOTLIGHT

✓ Better code

Cover Refactor improves the testability of your Java source code with zero additional development effort. This code is easier to test, maintain and enhance.

✓ More coverage

Cover Refactor increases unit test coverage by default. The act of making more code testable means Cover Core can automatically write more unit tests.

✓ Reduce regressions

Risk is the heart of the matter. More testable code and higher unit test coverage reduce the risk of regressions when code changes are made - especially in complex legacy applications.

✓ Focus effort

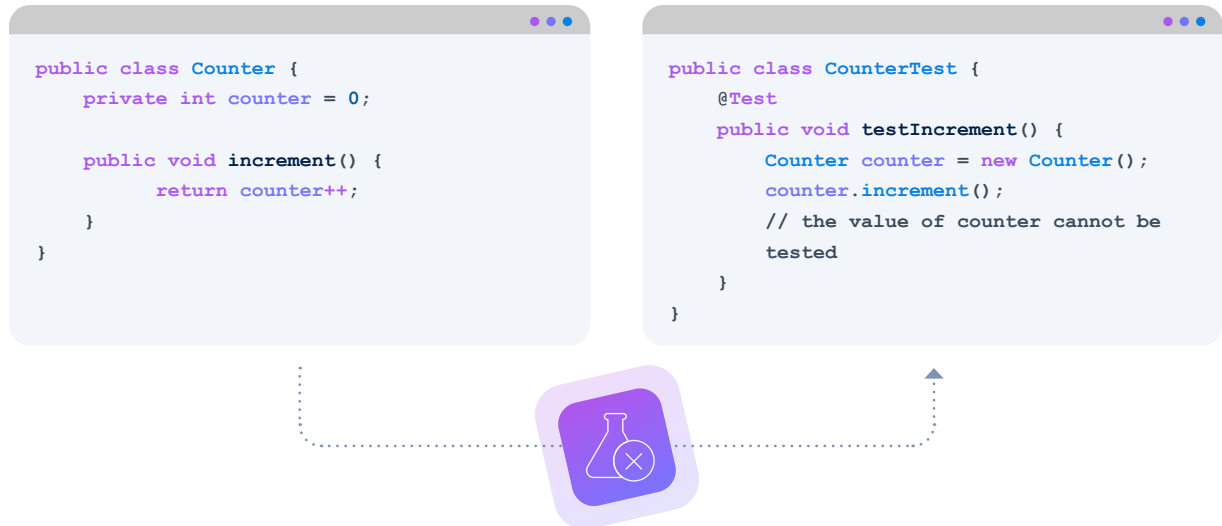
There's too much code to write and not enough people to do it, so it's hard to prioritize improvements to old code over productive new development. Refactor does some of the work for you.

✓ Understand testability

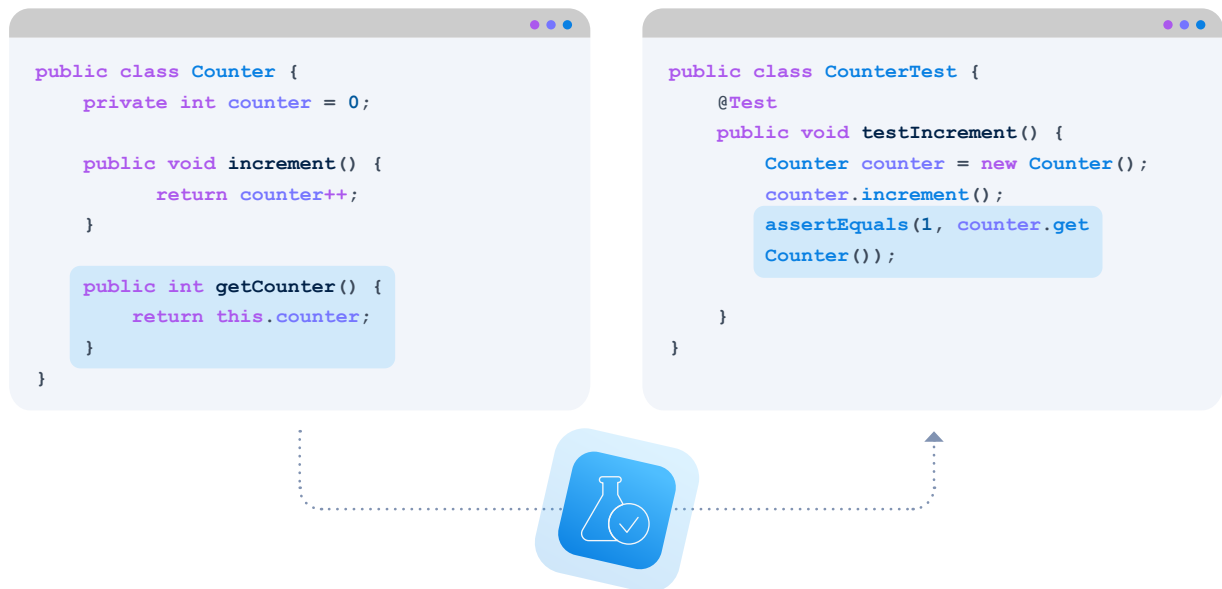
The analysis performed by Cover Core is the basis of Refactor's capabilities. Cover can visualize exactly how big the problem of untestable code is, and pinpoint specific issues to solve - no easy task in a large codebase.

Automatic Refactoring Example

This simple example shows how Cover Refactor can increase testability by improving observability. A method is testable when you can fully control the inputs and fully observe all of the side-effects of the method call. The private field `counter` is not observable by the unit test, so an effective test cannot be written.



Once Cover Refactor has updated the code in question, Cover Core can automatically write an effective unit test.



“Citi Markets uses its deep software expertise to move faster and be more competitive. We find value in Diffblue’s auto-generation of test cases. It helps drive test consistency and coverage of our software - freeing up developers to focus on delivering higher quality software, faster – and improves our developers’ experience”

Jonathan Lofthouse, Managing Director & Global Head of Markets Technology, Citi

Diffblue Cover helps you increase business agility and accelerate transformation



Optimize the velocity and quality of Java teams; catch regressions early

The Diffblue Cover Platform

Diffblue Cover includes a range of features that let you extract more value from Java unit testing:



Reduce software development costs and increase productivity



Untangle the complexity of refactoring legacy code



Accelerate modernization and cloud migration of core applications

ABOUT DIFFBLUE

Founded by leading computer scientists from the University of Oxford, Diffblue is changing the way code is developed. The company’s flagship developer tool, Diffblue Cover, uses AI to automatically write unit tests that help Java development teams and organizations deliver better, more modern software at higher speed. Diffblue: AI for Code. Learn more at [Diffblue.com](https://diffblue.com) or contact us at info@diffblue.com

© Copyright Diffblue Ltd 2022

