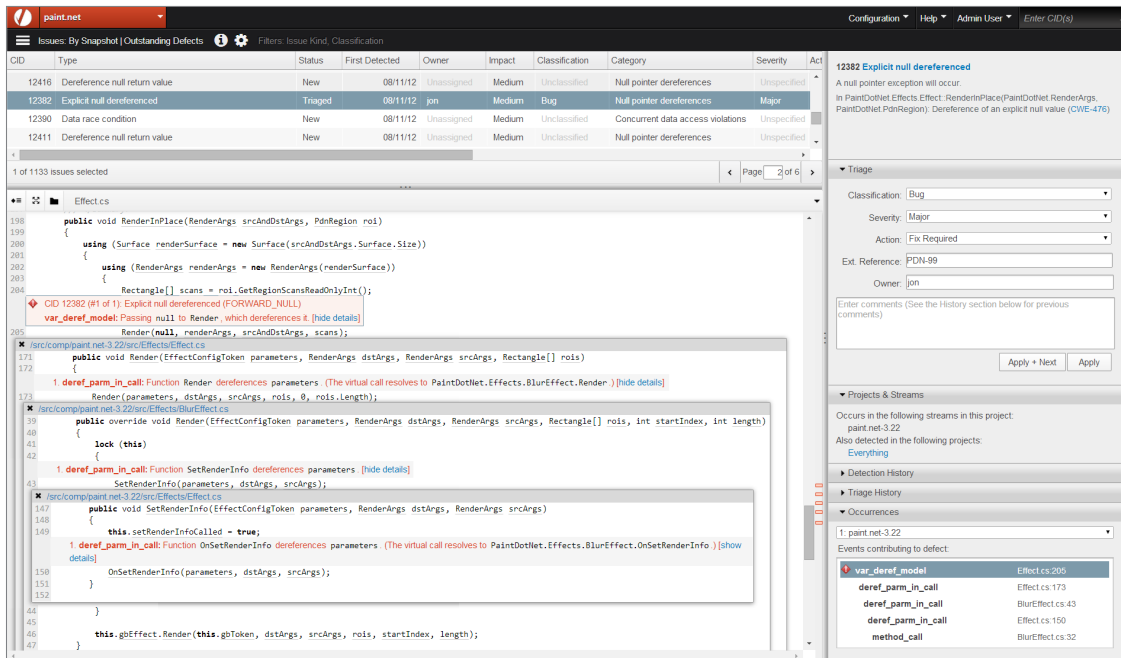


Coverity Software Testing for C#

The Coverity platform helps identify, manage and remediate critical defects in the code and improve the overall efficiency of testing efforts, reducing the cost, time and risk of software errors.



Find and fix C# defects from within Visual Studio, from the command line or as part of the centralized build.

Solution Overview

The Coverity Software Testing Platform allows developers to build testing into the development process at the earliest stage, and enables Quality Assurance (QA) teams to prioritize manual and automated testing efforts without requiring access to the source code. This reduces costs and risks, accelerates time-to-market, enhances customer satisfaction and increases revenues.

Key Capabilities

Deep Code Intelligence with Sophisticated Analysis Techniques

The Coverity Software Testing Platform applies multiple patented analysis techniques to automatically test code as it is written and accurately detect issues. For Development, the platform analyzes the source code rather than the binary code, providing a more complete understanding of the code. It provides full path coverage, ensuring that every line of code and potential execution path is tested. The platform's interprocedural analysis enables developers to easily find and fix complex defects that cross function boundaries.

The Coverity platform utilizes sophisticated techniques to minimize false positives and provides actionable remediation advice. Users can analyze large C# code bases quickly by leveraging parallel analysis. This means even the most complex code bases can be scanned regularly, which allows teams to adopt software testing as part of a nightly or continuous build process. Incremental analysis saves time by enabling developers to quickly reanalyze modified code—directly from their desktop or from within their IDE.

The Coverity platform also helps QA teams focus their automated and manual testing efforts based on change impact and without access to the source code.

IDENTIFY AND REMEDIATE CRITICAL C# ISSUES

- Concurrency defects such as deadlocks, race conditions and locking misuse
- Performance degradation problems due to memory leaks, file handle leaks, custom memory and network resource leaks, and database connection leaks
- Crash causing errors such as null dereferencing
- Incorrect program behavior caused by dead code or control flow errors
- Common arithmetic and typo issues such as copy/paste errors
- Missed automated tests in high-risk areas of the code
- Holes in manual and automated testing plans that could introduce regression risk

Manage and Remediate C# Issues

Coverity Connect is the collaborative issue management interface used by Development to efficiently manage all issues surfaced by the Coverity or third party analysis engines, from identification to resolution, within a unified workflow. Defects can be automatically assigned to the appropriate developer for resolution, and users can prioritize and filter issues based on criticality and impact. Coverity Connect provides source code navigation that helps developers understand the exact path to the defect and automatically identify every occurrence of the defect across shared code.

Identify Critical Defects

The Coverity platform identifies critical defects in the developer's workflow with accuracy and actionable remediation guidance. It enables developers to efficiently troubleshoot and fix the quality defects that matter, such as concurrency issues, null dereference errors and resource leaks, before the code goes to QA.

The analysis results are highly accurate so developers don't waste time managing a large volume of false positives. It provides actionable information and precise remediation guidance, showing the right way to fix the defect and the best place in the code to fix it. This effectively builds quality into the development lifecycle which reduces the cost, time and risk of software defects.

Analysis packs are available for seamless integration of additional analysis results into a unified workflow, providing a single location to triage and remediate defects. Some packs provide additional capabilities such as architectural analysis, while others provide integration with third party analysis tools like FxCop.

To make software testing a natural part of the SDLC process, the Coverity platform provides seamless bi-directional integration with many leading lifecycle tools. With the Microsoft Visual Studio IDE plug-in, developers can clean their code prior to checking it into a centralized build system while the code is fresh in their mind and easiest to fix. Defects that are identified through the central build can also be easily remediated from within the IDE—integrating software testing directly into their current process. The platform also integrates with other critical tools and systems such as source control management—including Microsoft Team Foundation Server, build and continuous integration, bug tracking and application lifecycle management solutions.

Improve Automated Testing

Automated testing can be effective but is often inefficient in practice. Code coverage is commonly used as the metric to

determine “adequate vs. inadequate” testing. While coverage tools provide basic guidance about the percentage of the code that's covered by an automated test, they lack the ability to understand the ripple effect of change or help teams prioritize their automated testing efforts.

Coverity Test Advisor – Development Edition enables teams to focus and prioritize their testing efforts on the most critical areas of the code. It provides the code intelligence required to establish and enforce testing policies that define what must be tested, such as all new code and legacy code impacted by a change, as well as what can be ignored, such as exception handling or debugging code. Users also receive guidance about which existing tests they should run based on the impact of change. Violations of the established policies can be automatically assigned to the appropriate team member for quick and efficient remediation, and stage gates can be implemented to validate when code has been adequately tested. With Coverity Test Advisor – Development Edition teams get better visibility into what tests they need to write and which tests they need to run, lowering the risk of software failures.

Coverity Test Advisor – QA Edition provides QA teams with intelligent change impact analysis for software testing. By monitoring the execution of applications and aggregating results of both manual and automated tests, teams can prioritize tests and identify which are most critical based on the changes to source code. The Impact Analyzer evaluates the impact of modifications to the test plan, allowing QA to eliminate redundant tests and generate scenarios that eliminate gaps in test coverage. This “what if” potential allows teams to model changes before the testing cycle begins and use that information to properly schedule risky changes to the product so they can be adequately tested.

Drive Adoption and Mitigate Risk

Coverity Policy Manager allows an organization to define and enforce a consistent standard for code quality, security and testing. It provides visibility into which teams, projects or components are compliant with these standards and can create measurable stage gates based on specific criteria regarding defects and testing.

The customizable views in Coverity Policy Manager enable teams to select development metrics and thresholds that align to their objectives, with the flexibility to modify them throughout the course of the project. Managers can monitor and pinpoint areas of risk by drilling down into specific issues. This code intelligence enables better decisions and improves the predictability of releases.