



Jtest

## Improve Developer Productivity

- » Analyze code to increase the team's confidence in code quality and security and comply with coding standards.
- » Accelerate Agile development and increase code coverage with fast and easy AI-powered unit test generation.
- » Efficiently test new and legacy code to locate defects that impact software security, quality, and reliability.
- » Automate regression testing through integration with CI/CD pipelines to view static analysis and coverage results.
- » Get immediate feedback on changed code in the IDE and CI/CD pipelines with test impact analysis.

## Optimize Java Development

Get a [free trial](#) to see how to improve the security, performance, and quality of Java applications.

# Parasoft Jtest

## Java Developer Productivity Solution

### Work Smarter With AI-Powered Automation

Parasoft Jtest accelerates Java software development with in-depth static analysis and AI-powered automated unit testing for the delivery of reliable, secure, and maintainable software. Maximize quality and minimize business risks while optimizing developer productivity. Seamlessly integrate Jtest into your IDE (IntelliJ, Eclipse, or VS Code) to support developer workflows or into the CI/CD pipeline for analysis and reporting.

Quickly analyze Java code for critical defects and security vulnerabilities with static code analysis. Leverage both preventative (pattern) and detection (flow-based) static analysis techniques, along with a comprehensive set of metrics for code structure. Get AI-based recommendations about prioritized violations to fix and the appropriate developers to assign to fix them. Integrate Jtest with OpenAI/Azure OpenAI providers and accelerate remediation of static analysis violations with AI-generated fixes. Validate code reliability through compliance checks for CERT, CWE, OWASP, PCI DSS, and more. Automatically generate compliance verification documentation.

Efficiently perform unit and integration testing with powerful automated unit test generation for new and modified code. Jtest's AI-led Unit Test Assistant automatically generates an optimized suite of meaningful and maintainable test cases with mocks and assertions. By supporting developer-assisted workflows, teams can efficiently build and configure additional unit tests to fill coverage gaps. Leverage Jtest's OpenAI/Azure OpenAI integration to analyze and refactor existing JUnit tests based on user-provided natural language prompts, enabling users to more easily and quickly modify unit tests based on their needs.

Get instant feedback about the quality of modified code using test impact analysis to run an optimized set of tests based on recent code changes. Parasoft Jtest's integrated Java development testing solution helps organizations reduce risks, cut costs, increase productivity, and achieve industry compliance goals.

Comprehensive and configurable reporting enables developers and managers to understand and prioritize errors detected in the codebase. Import static analysis and code coverage from CI/CD pipelines (GitLab, GitHub, and Azure DevOps) into the IDE. Part of the powerful Parasoft Continuous Quality Platform, integration with Parasoft DTP enables teams to get visibility into code coverage from all testing practices, including unit, API, UI, and manual testing. Additionally, measure code coverage against new and modified code to ensure sufficient coverage even if overall code coverage is low.

“Our developers are spending less time fixing production issues and more of that time writing code. In addition to that, as developers are writing more tests, hopefully they’re finding bugs sooner before they get into QA.” – Fitch Solutions

## Automate Code Analysis & Compliance Reporting

Shift left the prevention of security and quality defects with static code analysis and SAST. Parasoft Jtest quickly goes beyond the capabilities of open source solutions to fully audit code, finding issues ranging from API misuse to security flaws, helping ensure that the code adheres to industry standards.

Validate code reliability with compliance checks and deep code analysis. Evaluate code against 1000+ built-in static analysis rules for Java, including CERT, CWE, OWASP, PCI DSS, and other security standards, enabling early-stage detection of defects.

### Analyze & Optimize Code

Expose bugs without executing code. Get ahead of compliance requirements by integrating static code analysis directly into your dev workflow. Use flow analysis to simulate complex application execution paths and identify those that could trigger runtime defects. Analyze code metrics against defined thresholds to find out-of-range values.

### Customize Rules

Define organization-specific guidelines and coding standards. Create custom pattern-based rules specific to your development environment and practices.

### Eliminate Defects

Quickly find code defects with complete path analysis for accurate violation detection. Accelerate code fixes with Jtest's AI-generated recommendations. Get immediate feedback directly in the IDE.



### Optimize Issue Remediation With AI

Accelerate the triage of static analysis violations by leveraging AI to identify high-priority issues and automatically assign violations to the developer who is most fit to fix the violation based on past triage actions. Address violations faster by following AI-generated recommendations.

### Measure & Manage

With centralized reporting and AI-powered analytics, the team can prioritize errors detected in the codebase.

## Reduce Unit Testing Time & Effort By Half

Automate test creation with easy one-click actions for creating and maintaining meaningful JUnit tests. Jtest's Unit Test Assistant uses AI to automatically create JUnit tests, along with recommendations and quick fixes, to simplify test creation for new and modified code. Parasoft Jtest enables developers to optimize the time it takes to validate their code and helps to ensure that each part of the software performs as designed and with continuous quality.

### Generate Meaningful JUnit Tests

Bring AI and automation to the process of unit testing. Easily create, assess, and enhance JUnit tests with real-time, context-aware assistance, leaving developers free to focus on the business logic of the test.

### Create Unit Tests In Bulk

Quickly generate test suites that cover a broad range of the code base with bulk creation to significantly increase code coverage. Leverage existing JUnit test suites and expand them to cover more use cases with test case cloning and automatic mutation.

### Achieve Code Coverage Targets

Expand the test suite further with Jtest's developer-assisted workflows to achieve even higher coverage. Cover untested areas of code with quick fix and bulk actions to increase code coverage and jump start your test suite. Track coverage for modified code in addition to overall code coverage.

### Leverage OpenAI/Azure OpenAI Integration

Enhance unit tests by leveraging Jtest's integration with OpenAI/Azure OpenAI and the use of human-crafted natural language prompts to analyze and refactor existing tests based on specific requirements.

