



## CASE STUDY

# AI-Driven Java Unit Testing Boosts Developer Productivity for Financial Firm

## OVERVIEW

A leading multi-national financial services institution produces applications that enable their customers to access a variety of personal banking, mortgage, and loan services via the company website.



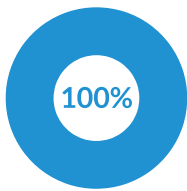
**Industry:**  
Financial

**Company Size:**  
42,000

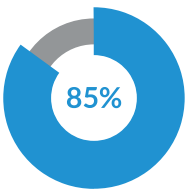
**Location:**  
United States/  
Global

**Solution:**  
Jtest

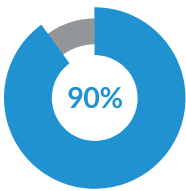
## KEY RESULTS



Accelerated unit test generation by 100%.



Achieved 85% code coverage in a few weeks.



Reduced test execution time in the CI/CD pipeline by 90%.

## THE CHALLENGE

Like many other financial institutions, DevOps was a key initiative at this leading financial services organization and the development teams were under pressure to accelerate software delivery. They had hundreds of Java microservices, many containing legacy code that had little code coverage. Evidence indicated that changes made to the services with lower code coverage correlated to significantly higher defect rates.

Developer productivity was another important initiative. Poor coverage was creating an alarming amount of late-cycle rework. Furthermore, developers were spending too much time trying to manually create unit tests to increase coverage. These challenges were creating unpredictability in both quality and delivery timelines. Software delivery leaders were looking for a way to help the developers work more efficiently to increase productivity without sacrificing quality.

## THE APPROACH

The organization determined that they needed to find a solution that would:

- » Measure existing code coverage levels for both automated and manual testing efforts.
- » Automate the generation of unit tests to maximize coverage.
- » Support the following tools and technologies.
  - » JUnit testing frameworks 4 and 5
  - » IntelliJ and Eclipse IDEs
  - » Maven and Gradle build systems
  - » Jenkins and GitHub CI systems
  - » Spring, Spring Boot, and Lombok

The organization faced difficulty in selecting a unit testing tool due to the market having multiple free and semi-automated tools. After researching for a few days and evaluating the options, they selected a few tools including Parasoft Jtest.

The company moved forward with a proof of concept with Jtest where Parasoft provided them with:

- » Excellent guidance and support for implementing the Java testing solution to run unit tests
- » Visibility and actionable metrics from an automated collection of existing code coverage levels
- » One-click generation of unit tests that increased coverage by three times in just a few minutes





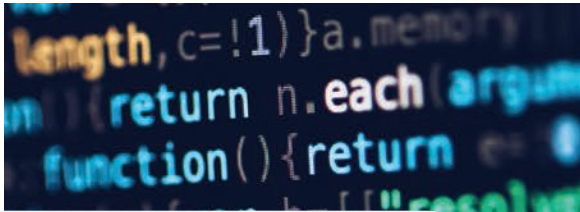
## THE SOLUTION

Independent research, several demonstrations, and a successful proof of concept made Parasoft Jtest the clear choice for the organization.

Using AI to automate unit test generation, Jtest offered some clearly differentiated capabilities that made it easy to build the business case.

- » **Unit Test Assistant.** Parasoft Jtest's AI-enabled Unit Test Assistant is an IDE plugin that helps guide users through the unit testing practice with easy one-click actions for creating, scaling, and maintaining unit tests.
- » **Test impact analysis.** Jtest leverages AI to selectively target only changed code and identify the right subset of tests to validate those changes.
- » **Spring and Spring Boot support.** Specialized support for these frameworks reduces a huge amount of effort for developers for MVC tests, mocking, and more.
- » **Aggregated coverage.** Developers can view code coverage markers within the code for combined code coverage based on multiple testing practices, like unit testing, manual testing, and functional testing. The holistic view focuses testing efforts, which increases efficiency.
- » **Static application security testing (SAST).** A huge bonus with Jtest is that it can also perform automated static analysis testing for code with support for CWE, OWASP, and more. While the company wasn't looking for SAST, they realized Parasoft offered a solution that met their need to test for security vulnerabilities.


After implementing Jtest, the development team at this financial organization was able to generate comprehensive test suites in a matter of hours. Within a few weeks, they reached their goal of 85% code coverage on modified code for their most business-critical microservices.



*“Parasoft Jtest has taught our engineers how to be better developers. In the process of using Parasoft Jtest, we’re getting fewer and fewer defects in the code base because developers are learning how to write better Java.”*

—Senior project lead engineer at the financial services organization

Since AI optimizes tests for uncovered code to boost existing coverage, the development team was able to create a lean test suite without redundant tests. All of this significantly reduced the burden on the team, enabling them to focus on other critical and innovative aspects of the development process. Jtest’s ability to execute tests only on the code that changed reduced the amount of time required for testing and eliminated unnecessary testing, which further increased productivity.



*“Test impact analysis examines our source code to find the changes a developer has made in the application and automatically identifies the subset of tests that need to be executed to validate the changes. The time savings is incredible!”*

—Senior project lead engineer at the financial services organization

## THE RESULTS

Using Parasoft's AI-enabled Jtest, the team was able to create better unit tests and drive higher code coverage. The solution delivered positive business outcomes for both of the following strategic initiatives at play in the organization.

1. **Increased developer productivity.** The Java developer productivity gain was estimated at 4-10 hours a week of time per developer that could be shifted from unit testing to innovation. That adds up to more than 20,000 hours annually for their team of 100 developers.
2. **Improved code quality.** Defect rates dropped by more than 20%, resulting in:
  - » Accelerated delivery
  - » Reduced rework
  - » Improved confidence in software deliveries

*"Since we implemented Parasoft Jtest, we have successfully reduced the amount of time it takes to create and maintain unit tests by more than 50%."*

—Director of development at the financial services organization

The organization was able to create better unit tests and drive higher code coverage with AI-powered unit testing. They achieved significant results that enabled them to rapidly deliver high-quality software.

- » **100% faster unit test creation.** One-click generation of unit tests for legacy code and services improved developer productivity. They built a comprehensive and meaningful suite of JUnit test cases two times faster.
- » **85% code coverage in a few weeks.** The company increased code coverage from 20% to 85% coverage on average for their most business-critical microservices.
- » **90% reduction in regression test execution time in their CI/CD pipeline.** Jtest leverages AI to detect code changes and only executes the tests that are impacted by change.

Parasoft's AI-powered automated Java testing solution integrates quality into the software development process by preventing, detecting, and remediating errors earlier in the SDLC. This significantly reduces the cost of fixing defects.

In addition to increasing the development team's productivity, this financial organization reduced the overall cost of testing. The ROI of implementing the tool was achieved in less than three months. They delivered quality code faster with fewer defects at a lower cost.

## TAKE THE NEXT STEP

[Start a free 14-day trial](#) of Parasoft Jtest with AI-enhanced automation to maximize developer productivity with fast and easy Java unit testing.

### ABOUT PARASOFT

Parasoft helps organizations continuously deliver quality software with its market-proven, integrated suite of automated software testing tools. Supporting the embedded, enterprise, and IoT markets, Parasoft's technologies reduce the time, effort, and cost of delivering secure, reliable, and compliant software by integrating everything from deep code analysis and unit testing to web UI and API testing, plus service virtualization and complete code coverage, into the delivery pipeline. Bringing all this together, Parasoft's award-winning reporting and analytics dashboard delivers a centralized view of quality enabling organizations to deliver with confidence and succeed in today's most strategic ecosystems and development initiatives—security, safety-critical, Agile, DevOps, and continuous testing.