

JUnit and LISA: The Perfect Match

by John Michelsen, Chief Architect and Founder, iTKO, Inc.

JUnit - The Standard for Test Execution

JUnit is an open source Java framework offered as an enabler for developers to do unit testing.

It has been adopted as a de facto standard in the Java world. Almost anyone developing in Java is using it. Since JUnit alone cannot support the entire development process, countless open and proprietary projects have been started to help close its test coverage gaps.

The Problem - Poor Productivity & Lockout

The unintended consequence of JUnit is that test creation has become a very labor-intensive programming task. Non-trivial functional tests and load testing are cumbersome coding tasks that provide little contextual feedback. And non-developers cannot participate.

The Principle - Efficient Test Creation

Software quality demands constant test creation. As many stakeholders as possible must be able to build the most comprehensive tests possible. If test creation is too expensive and difficult, you will give up on it. Efficient test creation is the enabler for quality.

JUnit + LISA: Best of Both Worlds

LISA's advanced testing capabilities; fast, no-code test creation; reporting and results can be rolled right into JUnit, and executed as JUnit test suites.

"I'm four times as productive with LISA as I am with JUnit alone." - Jon Caulfield, veteran enterprise software architect and Java guru

You can use the JUnit+LISA and Jakarta Ant infrastructure to perform not just unit tests, but any kind of LISA test. For example, you could add a LISA test to your existing JUnit/Ant build that launches a 10,000 user load test that fails the build if certain key performance metrics are not met. Tremendous LISA power, applied in an easy, pro-Java way.

Your JUnit tests can be executed and pulled together by LISA to be executed by non-developers such as QA analysts. LISA's powerful no-code Java execution capabilities allow the entire team to continuously test, while developers can still uncover exactly why their LISA tests pass or fail within the suite.

Buy vs. free? Or buy vs. build?

There are meaningful reasons to embrace open source initiatives. But when you are trying to test a complex environment, free testing alone is far from free. It takes considerable development cost, expertise and effort to ensure quality throughout the entire development cycle. LISA offers incredible ROI by keeping JUnit for developers, and adding regression, stress and load testing that integrates with every layer of a J2EE and service architecture right out of the box. For productivity, there's no comparison.

Compare testing with LISA vs. using open source alone.

Open source technologies can improve many aspects of your development process. However, if you value development time, and want to ensure that everyone on the team can become accountable for quality, take a closer look at adopting LISA.

Why JUnit users go with LISA

At iTKO, we use JUnit, and LISA fully supports and extends JUnit in many compelling ways. We are glad that the software market is better focused on testing, in part due to JUnit.

However, some teams view JUnit as the hammer and every testing need as the proverbial nail. Here are some of our concerns with a unit-test-only strategy:

1. Unit testing alone is simply not going to cut it for quality. Developers are notorious for 'happy path' testing. Developer testing won't find missed requirements and performance problems in the completed deployment. Mistakes hurt, but misunderstandings kill when it comes to software quality.
2. A test strategy that doesn't bring the QA/Business function in until the end is really no different than a waterfall, test-at-the-end process. It has all the same old issues we are trying to iterate our way out of.
3. JUnit requires all testing to be hand coded in Java, so your tests have bugs at the rate of 10-14 per KLOC*. And there's a huge productivity issue when you think about the number of lines of code required to sufficiently test a complex system.
4. XP/Unit Testing practices can help new development efforts, but what about all the existing systems you support? To support and test beyond the code level to existing EJBs, databases, websites and web services, you have to develop your own adapters or try to stitch together a bunch of other open-source components and point solutions. Which is not only costly, but invites more integration failures into your testing process.
5. JUnit is oriented around unit testing, but you need support for end-to-end functional, load, and production testing as well.

* per Thousand(K) Lines Of Code

Requirement	Open source contribution	LISA solution
Focused on developers, QA and Business Analyst team members	Almost exclusively focused on developers	Focused on every constituent in the development process - everyone should own quality
Standard way to report test results	JUnit/Ant provides a universally appropriate framework for pass/fail reporting	Expands JUnit's reach with non-developer functional testing, load testing and monitoring via that standard framework
Point-and-click "Acceptance testing" of completed systems	Some single-technology, code-based testing only	LISA provides a complete testing platform for acceptance testing of Web and Java Swing* applications
Build integration with an automated test framework	Ant and other related tools provide a de facto standard that JUnit supports nicely	LISA leverages this integration for build-time execution of unit, regression, end-to-end, and load tests
Integrated testing of multiple layers of a composite application	No single project with enough breadth, all code-based and separate processes	LISA is the industry's most comprehensive point-and-click middle-tier testing solution
Test process management	A number of projects provide some support for issue tracking and test execution	LISA integrates nicely with most test or process frameworks, open source or commercial
Performance testing	JMeter, OpenSTA, PushToTest, and other code-based tools can be used to some degree of load testing	LISA provides seamless migration of functional tests to load testing with all the features you'd expect from the industry leading platform
Production monitoring support	Few open-source tools are available	LISA provides the most relevant production monitoring needs like scheduled execution, metrics, thresholds, exceptions, email/sms generation. All with existing tests.
Point-and-click middleware testing, including Web Services, EJB, Java, messaging, etc.	Open source is almost universally code-based	LISA exclusively provides point-and-click testing of all these technologies in a single solution
Collection and reporting of SNMP, JMX and Windows Perfmon metrics	Some open source projects can be coded to collect some of these	LISA provides simple, live-interaction selection of all these metrics
Long-term maintenance and support pledge	Abandonment rate of open source is very high - more often new projects replace old ones, not evolve existing ones	iTKO provides on-going support and maintenance you would expect from an industry leader