

A BZ Media Publication

Software Test & Performance

VOLUME 4 • ISSUE 8 • AUGUST 2007 • \$8.95 • www.stpmag.com

**BEST
PRACTICES:
Testing With Eclipse**

A Tail of Two Test Teams

**Dogged Java EE
Unit Testing**

**Bark! Bark! Bark!
Protect User Roles
During Security Testing**

Sit! Heel! Debug!

***Training Your Unit-Test Team Takes
Time, Patience, Treats***





The Key to Testing With Eclipse? Community.

Eclipse is growing up. The upstart open-source, platform-independent software framework turns seven in just three months. That's old enough to have moved beyond vaporware and breathless hype into a slew of commercial products.

And it's not just IBM—which donated Eclipse to the open source community in 2001. Today, Adobe, Intel, Nokia, Sybase and scores of smaller companies ship Eclipse-based software.

As any parent can tell you, part of getting older is learning the basics of responsibility. You know, eat your vegetables before dessert, brush your teeth before bed, and so forth. Software developers may not necessarily follow this advice. But hygiene habits aside, good coders quickly learn that all successful programming projects, including those utilizing Eclipse, depend on sound testing practices. Among these: pay attention, no matter how easy it gets to practice automated, point-and-click development; define and publish clean APIs; and most important, always participate constructively in your community of peer developers.

Who better to discuss testing than members of the Eclipse community itself? After all, one of the wonderful things about the community, at least from a tech journalist's point of view, is that it's relatively open and accessible, provided you understand the basic norms of open source networks. Hint: authenticity and simple, concise questions are always preferable to pomposity and long-winded commentaries.

Researching this month's column, I drop into the eclipse.newcomer newsgroup at Eclipse.org to get some preliminary guidance on testing with the open source framework. One day after I



Geoff Koch

post my initial question, Mark Dexter chimes in with a pointer to a great screencam video on test-driven development with Eclipse. (Google "test first development using Eclipse, screencam" to find the video.)

"Eclipse has several features that help streamline [test-driven development,

such as] automatically creating empty methods [and] letting you run a program that contains compilation errors," writes Dexter, co-founder of a Seattle-based ERP company, Dexter+Chaney, that serves the construction industry.

"I'm very pleased with all the automation Eclipse allows, such as filling out try/catch blocks and suggesting method names as you type," adds Michael Donohue, the team lead for Java development at San Francisco-based Coverity, which released its Prevent Desktop for Eclipse product at the JavaOne Conference in May. "One gotcha is keeping your mind engaged. Since Eclipse does a pretty good job of making suggestions, it's easy to stop thinking about what comes out."

I could have researched this entire column by talking to small shops hanging out in Eclipse newsgroups. However, some of the biggest players in tech also contribute serious resources to the project, from programming talent to Website-infrastructure support. So it makes sense to also hear from a few of these more established players, several of which are placing big bets on their own Eclipse-based projects.

IBM Says API

With a clutch of products under its WebSphere, Lotus and Rational brands incorporating Eclipse, IBM is a logical first stop. What does Big Blue consider

to be the three most important focus areas for testing in Eclipse?

"API, API, API," says Harm Sluiman, an IBM Distinguished Engineer in Ontario, Canada.

Even down to the functionality provided by its core Rich Client Platform, Eclipse can be thought of as one kludged-together mass of plug-ins. This architecture is associated with several technical advantages, including a relative ease in extending Eclipse to non-Java languages, such as C and Python.

"Other integrated development environments have opened their APIs much later, and therefore don't have nearly as many open source plug-ins or commercial plug-ins extending their testing or other functionality," says Nada daVeiga, a Los Angeles-based Parasoft product manager for Jtest, available in its full functionality as an Eclipse plug-in.

However, the highly modular Eclipse architecture also puts a premium on defining clean APIs, which should only be done after strong collaboration with consumers, Sluiman believes.

"When components come together and the API usage is not clean, there is a high probability that there will be problems, especially from driver to driver, which is sure to disrupt any test effort," says Sluiman, who's on the project management committee of the Eclipse Test and Performance Tools Platform (TPTP) project. "Stabilization of a new API is a key milestone to track, and this should come before or early in the integration testing cycles."

Log It in Bugzilla, Stay Engaged

In its early days, Eclipse was an IBM-only experiment, but today, close to 20 com-

Geoff Koch writes about science and technology from Lansing, Mich. Write to him with your favorite acronym, tech or otherwise, at koch.geoff@gmail.com.



Best Practices

panies are signed on to the project as gold-plated supporters of the Eclipse Foundation. Behind their fancy title—the companies are declared to be Strategic Members—is a commitment to provide developer resources and membership dues that can be as high as US\$500,000 annually.

For that kind of dough, you might think these firms get special treatment. But that's not necessarily the case, says Bill Roth, vice president of Workshop products at BEA Systems.

“We are consumers of the Eclipse platform,” says Roth. “We perform customer acceptance testing of every build we take from Eclipse and then perform integration testing of our plug-ins.”

When BEA uncovers an Eclipse bug, the company's engineers log it in Bugzilla, just like anyone else in the Eclipse community. The key to getting the bug resolved, says BEA test engineer Rebecca Weinhold, is staying involved and providing feedback.

Weinhold uncovered squirrely behav-

ior in the Eclipse Web Tools Platform project while debugging JavaServer Pages in a BEA product. The Eclipse developer who responded to her Bugzilla report said the code appeared to be working as designed, but asked for Weinhold's opinion as to how it could be improved.

“The bug might've been dropped at that point if I hadn't responded with some feedback to make my case,” she says. “After a few exchanges, with comments from other Eclipse users, we came up with an improved user experience, which the developer agreed to implement and I promised to test.”

Weinhold's experience points out a last thought on Eclipse worth mentioning. Namely, as any child prodigy eventually learns, while it's great to be smart for your age, you eventually have to figure out how to work constructively with your peers.

Certainly, the expanding functionality of Eclipse and its pool of plug-ins is noteworthy. But more impressive is the coordinated teamwork displayed by the

Eclipse community.

“The main issues in software quality, like the toughest issues in software development, are process issues, not technical issues,” says Jeff Bocarsly, a Columbia Ph.D. and vice president of RTTS, a New York City-based software consulting firm. “The human problems of estimating the work effort required for a creative enterprise, organizing the work efficiently, communicating clearly between all stakeholders, reporting progress accurately, dealing with schedule and delivery adjustments during the work and documenting work properly are always the toughest to solve.”

Bocarsly's comments provide some context for the collective achievement of the Eclipse community: successfully delivering two major releases, Callisto and Europa, each with what Sluiman estimates to be tens of millions of lines of tested code.

Not bad for a seven-year-old. Now, just wait until Eclipse is old enough to start driving. ☒

Index to Advertisers

Software Test & Performance

Advertiser	URL	Page Number
Agitar	www.agitar.com/agile	6
Bredex	www.bredexsw.com	17
Eclipse World	www.eclipseworld.net	39
Hewlett-Packard	www.hp.com/go/software	40
IBM	www.ibm.com/takebackcontrol/innovate	20-21
iTKO	www.itko.com/lisa	25
Seapine Software	www.seapine.com/stptcm	4
Software Test & Performance	www.stpmag.com	8
Software Test & Performance Conference	www.stpcon.com	2-3
Software Test & Performance White Papers	www.stpmag.com	32
Test & QA Report	www.stpmag.com/tqa	31