Rethinking Software Testing

As organizations test for defects earlier in life cycle, developers are being trained to get involved

BY DAVID RUBINSTEIN

Writing tests for code in development can be a difficult process, dotted with potential pitfalls at almost every turn. The complexity of .NET and Java technologies, the use of Web services written outside the organization, language gaps between testers and developers as well as developers and business requirements writers, questions of source code versioning and even of when to test, all conspire to work against quality software.

Many development shops find testing to be such a problem that they put it off until the end, and often end up reducing the testing time to almost nothing when projects run longer than they should, according to vendors and development managers. The result—in far too many cases, they say—is software that doesn’t meet functional requirements, code that is loaded with bugs that developers find next to impossible to recreate, and, at the bottom line, a hit to a company’s profits.

In the early 1990s, testing tool vendors thought they had the problem licked when they came out with automated testing tools that could run unit and functional tests at regular times. But according to Dave Locke, program director in the Rational software division of IBM Corp., vendors pushing automated testing tools “shot themselves in the foot a bit. There were promises of point-and-click ease and ‘it’ll be amazing.’ It wasn’t.”

The director of quality assurance at a major cruise line company, who did not wish to be identified for this article, said, “We’re not a shining example of the benefits of automated QA. Part of it is skill set. Our existing testing people are not strong in automated testing. And I’m still not convinced of the value of automated testing for a Web site or things that change quickly.”

Automated testing does have its benefits, others argue. Regular builds and testing can help improve the development process, they say. “If you’re building every night and running tests, you find out right away if the software does what it is supposed to,” said Robert Leahey, director of developer relations for test tools vendor Automat edQA Corp. “There’s no I’ll take care of that in a few weeks.” It’s a dramatic change in the way people code.

PROCESS MAKES PERFECT

Indeed, many advocates of strong development processes believe testing earlier in the cycle is critical to creating high-quality software. One of the key principles of the Extreme Programming methodology, as detailed in the book “Extreme Programming Explained” by Kent Beck, for example, is that it relies “on automated tests written by programmers and customers to monitor the progress of development, to allow the system to evolve, and to catch defects early.”

While vendors and industry experts maintain that more testing is being moved up in the development cycle, at least one development organization is going in the other direction. “We’re moving away from writing code and testing at the same time,” said Bob Armstrong, director of Internet and Information Systems at Delaware North Co., a holding company for several subsidiaries involved in the hospitality and food services industries. “If I’m doing QA and development on the same servers, it’s hard to know where you’re at. You can’t do consistent regression testing if things are changing midstream.”

The company’s policy is to maintain separate development and QA areas, and it has been able to reduce the costs of hardware and software by using virtualization software from VMware Inc. to “clone” the development environment for testers. Armstrong said this reduces DLI conflicts and versioning problems.

There is wide agreement, though, that catching defects earlier in the development cycle can save countless hours of development time, lower the cost of producing the software and reduce time-to-market. Some say it is the business side of a company that’s driving the push toward software testing.

“It’s not so much about the code as it is about the business process,” said Lori Gipp, vice president of marketing at automated testing tool vendor Solstice Software Inc., claiming that companies are spending more of their development budgets on integration projects rather than creating new code. “Very little is about changing code; it’s putting pieces together,” she said. “This is a testing exercise.”

The Y2K problem at the turn of the century put more eyes on the problem, Gipp suggested. “Post-Y2K, more business people got involved in validating and testing systems. It brought a focus on QA earlier, and the business influence [of a company] has a lot to do with where they sit in the spectrum of testing.”

Joe Oddo, Solstice president and...continued on page 30
Industry’s First Defect Tracking System is still the Best

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
REPRODUCED WITH PERMISSION OF THE COPYRIGHT OWNER. FURTHER REPRODUCTION PROHIBITED WITHOUT PERMISSION.