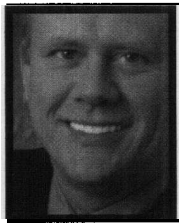


YES While many people are quick to blame the vendors, many factors outside a vendor's control can contribute to poor software quality By Bob Parker, AMR Research

ARE VENDORS DOING ENOUGH TO IMPROVE SOFTWARE?

I'm frequently involved in discussions about software quality. Enterprise clients share horror stories of upgrades and implementations gone wrong, including disturbing accounts of costs resulting from poorly engineered code. It's a strange position for me then to defend the quality of enterprise software—but it's one I welcome.



Too often, the blame is placed on vendors alone. On a closer look, several factors contribute to poor software quality, and many are outside the vendor's control. Among these are complexity, market conditions, a lack of standards, and overcustomization.

- **Complex construction:** A colleague likens the building of enterprise-class software to building a skyscraper. Large construction projects always come with quality issues and nothing is paid for until the "punch lists"—items that have to be fixed before the job is complete—are cleared. A major enterprise application, such as SAP R/3, will have more than 100 million lines of code. At 99% accuracy, there would be a million lines of bad code. That doesn't sound very good until you consider that there are usually only several thousand bugs (you can see the headline—"Thousands of Bugs Plague SAP") being addressed, so the quality is actually greater than 99.99%. Given the complexity of what's being delivered, most enterprise software vendors do an impressive job.

- **Conditioned conflict:** Suppose a builder, to assure business viability, had to go back to customers every year and sell them on major renovations. Or suppose the builder had to convince customers to raze their buildings and rebuild from scratch every five to 10 years. You'd have to conclude that

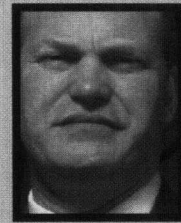
PHOTO OF BILL GUTTMAN BY BILL GRAMER

SQUARE OFF

NO The industry practice of rushing software to market and then patching bugs and security later is costing users a bundle By William Guttman, Sustainable Computing Consortium

A little less than 60 years ago, IBM founder Thomas Watson Sr., said: "I think there is a world market for about five computers."

As late as 1977, Ken Olsen, president and founder of Digital Equipment Corp., said: "There is no reason anyone would want a computer in their home."



I relate these quotes not to pick on anyone, but to point out that in the computer industry's formative years, even its visionaries didn't think we'd be as dependent on computers as we are today.

Obviously, we now live in a digital world. Our nation and our economy rely on effective and efficient data transfer. Information technology pervades our growing knowledge society. That's why software quality is so critical. Technology has come far, yet there's still a long way to go to achieve the software reliability, dependability, and quality we need.

The dimensions of the problem are huge: According to the Motorola Software Group, defective software costs global business an estimated \$175 billion per year. Another study attributes 45% of computer downtime to software flaws. With more than 110 million computers online, problems from defective software have dramatic consequences for the performance and competitiveness of every business.

An April survey of IT-industry professionals by *InformationWeek* Research summed up the state of the industry: Fully 97% of the 800 managers surveyed reported software flaws in their systems in the past year. More than 90% blamed faulty software for lost revenue or higher costs. Some 62% said they believed the software industry was

something was wrong with that market. Yet enterprise software markets are constructed in this fashion. Vendors must constantly offer new extensions and rush delivery to meet customer and Wall Street expectations. The perpetual-license model must change—to higher initial prices, renewable licenses, or bigger annual fees—before serious quality improvements can be expected.

• **Standard standards:** In construction, local building codes occasionally conflict with national standards, but for the most part, builders can depend on a consistent set of rules. At AMR, we like to say the nice thing about software-technology standards is that there are so many to choose from. Multiple standards bodies propose conflicting approaches while ad hoc standards from infrastructure vendors further cloud the picture. Enterprise-software vendors tell me they maintain hundreds of test environments to try to accommodate all the possible combinations of operating systems, databases, and application servers.

• **Custom craziness:** Larry Ellison took a lot of heat for suggesting that the best way for users to assure quality would be to take Oracle applications as they come and not attempt to customize them. While Oracle has softened its stance, the chairman had a point. Enterprise apps should provide a solid foundation for integration, but heavy modification by customers only exacerbates quality problems.

• **Sharing responsibility:** A cynical observer might say this is just another excuse for not doing the job right. Criticism of software vendors for not taking appropriate quality measures is well founded, but improvements won't be forthcoming until fundamental changes are made in the way software is bought, implemented, and used.

In 2001, AMR published reports on enterprise-commerce management. We produced a vendor-expectations document that listed vendor requirements. Ten vendors have submitted disclosures so far, but many more are in the queue.

Vendors want to respond to users' needs. If users base purchase decisions on quality and other key issues instead of vision and marketing fluff, the market will serve as a powerful change agent for quality improvement. ○

BOB PARKER is VP and general manager of enabling technologies at AMR Research, a Boston analyst research firm specializing in enterprise software applications. He can be reached at bparker@amrresearch.com.

Nothing will
improve
until we
change how
software is
bought and
used



doing a bad job of producing bug-free software.

The software industry acknowledges that its current model—bringing new products to market to gain a competitive advantage, and patching bugs and security flaws after the fact—doesn't work. The result of this practice has been a crisis of confidence in the industry, and developers are on the front line. A single piece of software can have millions of lines of code. All it takes is three or four errors per thousand lines of code to lead to unexpected—and usually undesired—results. And with software embedded in everything from our telephones to our cars, toys, TVs, planes, and national-security systems, we clearly need to do better.

We need to make software as reliable as water or electricity, but we know there's no quick fix.

At Carnegie Mellon University we have formed the Sustainable Computing Consortium as a neutral ground for the discussion of how to develop computing that's sustainable, secure, dependable, and of high quality. The SCC has several objectives:

- To develop rigorous and valid methods for measuring and assuring sustainability.
- To develop empirical frameworks for analyzing public policies and their impact on sustainable computing.
- Tools to help assess sustainability and manage risks.

Measurement is essential.

Claims of software reliability are mostly subjective, and it's difficult to compare the reliability of products, whether open-source or proprietary. The SCC will develop reliability metrics to improve the quality, dependability, and security of software.

Better development standards are also essential. We know it's possible, and preferable, to build better code from the ground up, rather than attempting to fix design flaws, security, and interoperability problems once programs are released. The goal is to quantify those best practices and make them more widely available, changing the model from "release and patch" to "develop, test, fix, and release."

Vendors have a lot of work to do, but we're too dependent on software to ignore this industry's growth pains. ○

WILLIAM GUTTMAN is director of the Sustainable Computing Consortium at Carnegie Mellon University. *InformationWeek* is a charter member of the SCC.

Tell us your view at optimizemag.com/forum/squareoff