

Feature

Achieving Business Advantage through the Elusive Balance of Development & Support

By Eldad Maniv, VP of Product Management, BMC Software

There's no debate that software errors, or "bugs," are costly; we've all seen reports about how much money is lost each hour that a critical banking or retail application goes offline. But when software is your business, the impact of application problems can be devastating — any seemingly small issue that is uncovered during testing or is escalated from support that keeps engineering teams from focusing on development can have a costly ripple effect:

- Product release dates slip and new, competitive functionality isn't being developed
- Rushed testing cycles result in products being released with sub-par quality
- Skyrocketing costs for resolving complex customer support issues
- Increased competition leads to longer sales cycles and higher cost of sales
- Declining maintenance renewal rates from disappointed customers
- Damaged company reputation as support and functionality fail to meet customer expectations



**Eldad Maniv, VP of
Product Management,
BMC Software**

According to industry research, on average, developers spend 39 percent of their time finding and resolving problems in applications that are already live [1] and during testing, that jumps to nearly 100 percent - that may sound like an exaggeration, but think about your QA process: it's *all* about finding and fixing problems, and the bulk of that time is root-cause analysis (approx. 80 percent is finding the root cause and only 20 percent is correcting the issue).

Any executive in charge of development should be concerned by these figures. And *every* executive in a software business should be alarmed. It means that a significant percent of the time, your engineering team is not working on the next release, module, or product that will keep your maintenance revenues flowing, drive new sales, and maintain your competitive lead.

Optimizing Development Organizations

Hundreds of top Independent Software Vendors (ISVs) are successfully addressing these issues and are optimizing their development organizations by using application problem resolution technology to automate the traditionally time-consuming, iterative process of root-cause analysis. The technology has been proven to accelerate problem resolution times by up to 70 percent in both the development/testing and maintenance phases of the product lifecycle.

With problem resolution technology, ISVs gain a best practice approach to root-cause analysis that enables engineering teams to reclaim up to 40 percent of development bandwidth that is currently diverted from developing new products to supporting existing ones, which in turn, helps software vendors accelerate the launch of new releases, improve product quality, reduce support costs, improve customer satisfaction ratings, and meet service level agreements (SLAs).

Increased Complexity Means Increased Time to Resolution

In its 2006 Benchmark Study, the Service and Support Professionals Association (SSPA) reported that the percent of support cases closed at first contact continues to decline, while the length of time a case is open continues to increase (see Figure 1). [2] This too, is an alarming finding. Escalation to a broader resolution team leads to skyrocketing costs and takes engineers away from their primary development tasks. Longer problem resolution times also leads to unhappy customers and possibly delayed or lost revenues.

Service Levels Decline

Metric	2003	2006	Net Change %
Resolved on first interaction — Phone	54%	46%	-15%
Resolved on first interaction — Email and Web self-service	52%	40%	-23%
Average percent of issues resolved within 24 hours — Phone	63%	58%	-8%
Average percent of issues resolved within 24 hours — Email and Web self-service	59%	50%	-15%
Successful visits to self-service site	48%	44%	-8%

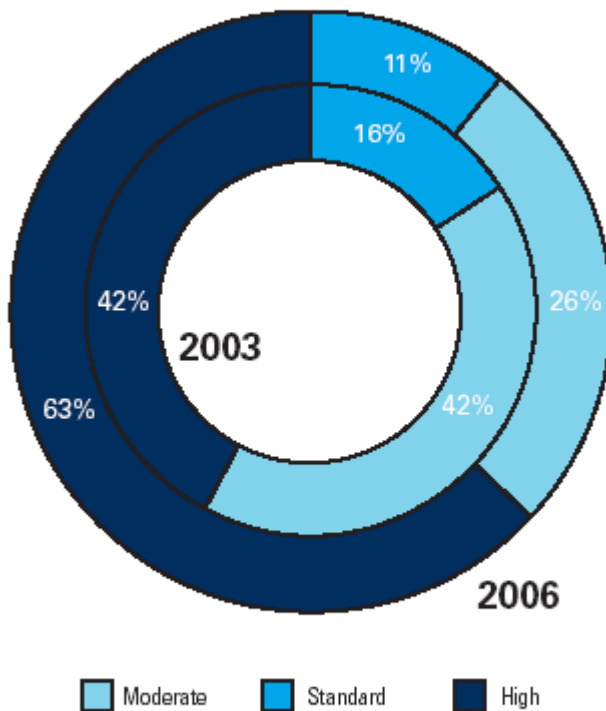
Source: 2006 SSPA Benchmark Survey Data

Figure 1. The percent of support cases closed at first contact is declining, while the length of time a case is open is increasing.

The characteristics that make software development and deployment methodologies, such as SOA and Web Services, so powerful and productive when all goes well, make problems extremely difficult to diagnose when things go wrong. Complex issues involving a number of component vendors and no obvious root cause can lead to unproductive finger-pointing, while customers wait impatiently for a resolution.

SSPA benchmarking research has also found that these difficult multi-vendor issues typically take four times longer to resolve [3], which only multiplies the potential negative impact on your business. A few years ago, the answer to longer support cycles and higher costs was a move to offshoring, Web self-service, and better knowledge bases for service desk staff. However, as figure 2 shows, these quick fixes have not been able to keep up with the rate of technology change and the increasing complexity of the software environment. This has led to an overall increase in the time development engineers spend on product support instead of development.

Complexity of Products Supported 2003 vs. 2006



Source: 2006 SSPA Benchmark Survey Data

Figure 2. Software environments continue to grow increasingly complex.

While the software development process has undergone multiple paradigm shifts in the past few decades, the problem resolution process continues to rely basically on the same manual, labor-intensive, iterative approach, which robs the engineering team of precious development cycles and contributes to those major productivity losses mentioned above.

Clearly, a change is long overdue.

Application Problem Resolution technology

A growing number of innovative software vendors are leveraging low-impact application recording technologies as a mechanism to capture the information needed for problem resolution – both during the testing process and for remote customer support. These software technologies look inside a running application to collect the technical information needed by engineers and support staff to determine the root cause of a problem. The more sophisticated technologies can capture not only the user's actions preceding a failure, but also the system configuration, performance parameters, and even the related code execution. They can also synchronize this information on a single timeline for further analysis. The most advanced systems provide users with powerful automated analysis capabilities to dramatically accelerate root-cause determination and problem resolution.

As any software veteran knows, the challenge is that the symptoms of a software problem rarely reflect the root cause. A single business transaction may kick off a sequence of complex processes, each of which may involve events that happen on up to a dozen potential servers. The root-cause of the problem could be a coding error, a hardware fault, a configuration issue, or even an end-user's mistake.

A recent survey, conducted by Dynamic Markets Ltd., found that 75 percent of application problem resolution cycle time is attributed to determining the root-cause of the problem.[4] Pinpointing the root-cause of application problems can be difficult when problems surface outside the development environment, and especially when they occur at remote customer sites. Development and Support team members typically go through a lengthy and costly process that includes endless conference calls, iterative attempts to gather information, costly trips to the customer site, and multiple attempts to recreate the problem scenario. And in many cases, it turns out that the root-cause was actually related to another vendor's product. Application problem resolution technology alleviates this painful process and eliminates time-wasting finger-pointing among vendors.

Conclusion

No company faces a greater business risk from software problems than a software vendor. To mitigate that risk and increase efficiency, ISVs should consider using application problem resolution technology to optimize the use of engineering resources and achieve that critical balance between delivering new products and supporting current ones.

As V.P. of Product Management at BMC Software, Eldad Maniv is responsible for BMC products that are sold into development organizations, including the BMC AppSight Application Problem Resolution System, as well as the business unit's professional services and marketing functions. Maniv joined BMC after two years in a similar role at Identify Software (creators of AppSight), which BMC acquired in May 2006.

[1] "How Visible are IT Problems, Really?" Dynamic Markets Ltd., Sept. 2004, p. 5.

[2] The Service and Support Professionals Assoc. (SSPA) 2006 Benchmark Study Report, John Ragsdale, VP, SSPA Research

[3] Ibid

[4] "How Visible are IT Problems, Really?" Dynamic Markets Ltd., Sept. 2004, p. 6.