



August 7, 2014

Taking a Stance on the Cost of Poor Quality Software

<http://www.devx.com/enterprise/taking-a-stance-on-the-cost-of-poor-quality-software.html>

By Scott Aziz

Development project failures and the subsequent accompanying re-work are well-known issues in our industry. It's come to the point where many have given up on trying to remediate the situation. However, the Cost of Poor Quality (CoPQ) is largely a controllable entity. The truth is that there is a general lack of concern for CoPQ in the development and testing communities, largely due to the hidden costs of poor software quality. We have to ask ourselves, why should we care about CoPQ? It's an accounting problem, right? Isn't this just a cost of doing business? The answer is, no. Or, more accurately — it doesn't have to be.

The fact of the matter is that poor software quality costs impair each of us, and are too large to ignore. Software development is a complex process, with many people involved, numerous requirements and plenty of moving parts. As an engineering discipline, we can do better. Much better. Each of these issues can be addressed and significantly improved with better practices and improved enforcement.

Poor Testing Practices

One of the key failure points exists at the testing phase. Along with the development community, the software testing profession must also evolve. We can do this by increasing preventative activities. On many occasions, the productivity of testing decreases when more defects are introduced. This is due to engineering churn. In this case, the cost of testing is higher and the time to complete testing increases.

Another practice that must improve is the casual nature of testing. Multiple studies show that 85 percent of defects found in production could have been detected by simply testing all possible pairs of values. The testing industry can additionally improve by changing its hiring practices. All too often, software testing is conducted by untrained personnel. These "low cost" employees actually increase the overall cost of development when critical flaws go undetected. Let's raise the bar and insist on using only certified test professionals.

In some cases, the CoPQ extends far beyond the development budget. This is a worst-case scenario — when loss of business and litigation become a factor. In one instance, a highly regarded investment firm was required to pay \$217 million to customers in compensation for a "significant error" in the code of one of its investment models. Another organization in the financial services industry was obligated to establish a \$62 million compensation fund for a

software failure. The bug, which only occurred during heavy trading volume, went undetected by testers.

In the future, the U.S. government may step in and raise the bar for software testing if the industry refuses to self-correct the situation and improve standards. The SEC is considering writing regulations that would require trading firms and other market participants to disclose issues with their trading programs and test them before they are used on the open market.

Re-work Drives Cost Increases

The CoPQ is often described in two categories; Internal and External failures. Internal Failure costs are incurred to fix quality issues before delivery to the customer. Loss occurs from churn and the opportunity cost of delayed delivery. External Failure costs are created after delivery to the customer. These costs are driven up by re-work, customer support, loss of business and sometimes even litigation.

When faulty software is delivered to the (internal or external) client, it triggers re-work. The effort needed to remediate the situation is not insignificant — in fact it's often the largest driver in development costs. In most cases, re-work can account for 40-50 percent of the overall cost of the project. This is a restrictive business model that can be improved.

The Road Ahead

There are a number of ways that our industry can take control, improve overall software quality and reduce development costs. The primary method is to shift it from the uncontrollable Failure Costs to the controllable Prevention/Appraisal Costs. With each incremental increase in Appraisal activities, such as reviews, we can expect a corresponding and larger reduction in our failure activities.

One of the biggest hurdles is fostering an environment of change. Change management is difficult for many, especially when a price tag is attached. Investments are needed to shift the majority of the Cost of Quality (CoQ) to the prevention and appraisal side of the equation. CoQ will not only be reduced significantly, but it will also be more predictable and more manageable.

The necessary changes also require open minds. Inspections are not the most enjoyable engineering task, compared to the creative and intellectually rewarding processes of designing and coding. A cultural change is needed as much as a change in business practices. Many programmers are possessive about their code. Inspections are labor-intensive and low tech - and sometimes insulting to developers - but one significant catch can save months of work, millions of dollars and significant reputational damage to the organization.

We, as software professionals and business leaders, must take a leadership stance. It's time to manage IT as a business within a business. This will directly help our departments and the bottom lines of our organizations. A focus on enhanced software quality leads to business growth, enabled through lower costs. It begins with a commitment to build the best product for the lowest cost to best compete in the marketplace. And the reduction of costs is not the only benefit of quality software. We can also improve time to market and increase customer delight.

About the Author

Scott Aziz is Vice President & Business Leader QA & Testing Services, Specialized Market Unit, at [Mphasis](#)