Speed vs Quality

How Fast Can You Deploy?
Introduction

You know that old saying we used to see in service stations: Fast Good Cheap—Pick 2. A lot of people think it applies to application development too, but does it?

For many businesses, “speed to market” or the ability to deploy the latest service or technology ahead of any competitors and with high quality, is a game-changer. And losers may quickly find themselves out of business.

Recognition of that fact has been the impetus for many new technology products, tools, and processes. Old school thinking is preventing many companies from taking advantage of these new developments, so let’s debunk some of the myths that prevent adoption of best practices for firms that depend on the fast deployment of a new functionality:

Myth # 1
*Automation is great for regression testing but it’s too expensive to create and maintain before a system stabilizes—usually some months after initial deployment.*

Truth: The root cause of most defects is in the requirements—requirements that lack clarity can be interpreted in multiple ways. Adopting “shift left” techniques to reduce defect creation and identify defects early in the development process also paves the way for employing automation early in the development lifecycle.

Myth # 2
*Well-written manual test cases form the repository from which automated tests can be created.*

Truth: The days of “capture/replay” automation are long gone. Automation tools such as Selenium enable programmatic automation, and clearly written test scenarios (without the detailed steps of a test case) are all that is needed to create great automation scripts.

Creating step-by-step manual test cases is a very time-consuming process. With the latest automation tools, automated testing is more easily developed and maintained. With the right approach, automation can be used even for unit testing—reserving step-by-step test cases for manual tests, which is typically less than 1/3 of an application’s overall testing.

Myth # 3
*QA and Testing is just a fancy name for testing.*

Truth: Though most people use “QA” and “Testing” interchangeably, they are two distinctly different functions. “Quality Assurance” refers to the work that is done prior to code development for the purpose of defect prevention. Testing refers to the work that is done after code development for the purpose of defect detection.

Too many project teams rush to create code when taking a little extra time for defect prevention would be much cheaper and faster in the overall development lifecycle process.

Myth # 4
*The business users are changing their requirements so fast that we cannot keep up with*
the backlog and still meet new development requests.

Truth: Numerous studies have shown that production defects and change requests most often have their root cause in poorly defined or ambiguous requirements. The process looks something like this:

Business users and business analysts work to define requirements for the system until they are confident that they are in mutual understanding. But too often they are envisioning common terms through their own filters and experience. Happening frequently, this also means the developer having a totally different interpretation—and that is the interpretation under which the code will be developed. It may have little to do with the actual business needs.

If the variance is uncovered in UAT, and does not represent anything terribly serious while workarounds are possible, it’s quite likely that impending deadlines will cause the team to decide to move forward and worry about the changes later. When the workarounds become unmanageable, requiring more effort than the time available, then the users start making change requests, often hiring additional help until IT can respond.

The problem perpetuates and we see this often in quality assessments. Until one digs deeper into the analysis and root causes of certain conditions, it is not readily apparent that a failure to focus on defect prevention has created a large backlog, which in turn impacts the IT team’s ability to respond to requests for new development/new functionality.

Conclusion
Today it is possible to have agility, speed, and quality in the system development lifecycle process. It requires the right tools, state-of-the-art automation in testing and deployments, a mature testing organization, and good process discipline to avoid the rush to code at the outset to ensure that the development team can implement great Quality Assurance techniques.

About the Author
Vinai Gangireddy leads the Enterprise Quality Assurance and Testing Practice (EQAT) at Ciber Global with over 20 years of testing experience. He has effectively executed multiple software testing projects. Working in different technology stacks and domains, he has assisted numerous clients in implementing quality best practices and testing solutions that have resulted in high-quality software.