

Software Quality Professional

VOLUME NINE • ISSUE ONE • DECEMBER 2006

Secure Software Engineering Processes: Improving
the Software Development Life Cycle to Combat
Vulnerability

Thorsten Schneider

Design and Analysis of Software Experiments

*Thomas F. Griffin III, David C. Ang, Thomas Lucy-Bouler,
Nancy Davidson, and Robert Gehling*

Are We Doing Good, Or Are We Doing Bad?

Linda Westfall

A Study of User Requirements for COTS: Computer
Off-the-Shelf Software Solutions

*Mansoor A. Al A'ali, Seema A. Al-Mahmood,
and Ala Al-Zobaidie*



ASQ

AMERICAN SOCIETY
FOR QUALITY™

www.asq.org

Top Five Quality Assurance Pitfalls: Traps to Avoid When Implementing QA Processes

By Shaun Bradshaw, Questcon Technologies

The software industry has seen a positive shift in the number of companies making quality assurance (QA) and testing a top priority. Organizations are beginning to realize that proper techniques not only reduce costs, but also provide a competitive edge in the marketplace as clients demand software free of errors. Companies seeking to implement QA processes often find it daunting to develop and implement a strategy from scratch and that it is normal to run into various challenges along the way. This article discusses some of the common problems organizations often face when implementing QA processes.

Lack of QA Strategy

Lack of a QA strategy is the most common issue companies experience when implementing QA processes, and it is also one of the most difficult from which to recover. Most organizations will not build a software product without a clear strategy and a solid plan, yet QA process implementation is often an afterthought.

It is common to see an organization release the third major patch to correct the functional issues, performance problems, and security holes found after the initial production release of a new application. Users are frustrated because the system doesn't work, management is upset with poor performance that tarnishes the company's reputation, and project team members are still trying to figure out why defects keep popping up. With everyone pointing fingers in frustration, the organization is prompted to take measures to improve software quality, which may unfortunately be as close to a "strategy" as the organization gets. To create software that performs as expected, reduces software development life cycle (SDLC) costs, and mitigates the risks associated with poor quality software, establishing a QA process should be viewed as a strategic initiative with clear goals and attainable milestones.

To establish QA processes strategically, initiatives must be viewed as a long-term project. It is usually best to start a project with an evaluation of the following areas:

- Organizational
- Project management
- Software development
- Software testing

For each of these four areas, identify issues and gaps that prevent teams from developing better quality software. Next, identify the key areas for improvement and develop recommendations for improving them. Write recommendations in a way that makes it easy to establish goals and measure the success of implementation. For example, the assessment may find that a lack of communication between the marketing

department and the technical infrastructure group leads to system performance problems. One might write a recommendation as, "Improve communication between marketing and technical infrastructure," which cannot be measured objectively. Instead, the recommendation should be, "Develop and implement a needs-analysis worksheet for the technical infrastructure group to be filled out by the marketing department during the analysis phase of all projects." The latter recommendation provides measurable goals to achieve.

As a next step, prioritize implementation recommendations. Some recommendations may be easy to implement, but don't provide a significant return on investment. It is helpful to perform a cost/benefit analysis on the major recommendations to determine which ones should be implemented immediately and which ones should be held until later in the strategic initiative.

Lack of Distinction Between QA and Testing

QA and testing are not the same concept. It is easy to find organizations where management believes better QA is needed, which in turn is translated as the company needs more testing. Unlike a lack of QA strategy, failure to distinguish between QA and testing is less detrimental to an organization's ability to ultimately implement QA processes. This is primarily because testing still needs to be performed; it just shouldn't serve as a substitute for QA.

QA is a continuous process of activities that ensure the application being developed is fit for use. On the other hand, testing is used to identify and remove existing defects, and measure the success of other QA activities. Testing is a subset of the overall QA process. Put another way, QA processes strive to prevent defects, while testing activities strive to find defects. For example, one common QA process is establishing coding standards. By having defined coding standards, developers trained in the use of those standards are less likely to make certain coding errors. This prevents certain performance problems or functional defects from occurring in the application.

As long as the organization clearly understands the difference between QA and testing, it may be easier to first establish strong testing processes using results and metrics derived from testing to help identify trouble spots or the "root cause" of defects found in the application. With the proper QA strategy, having a strong test process can drive better QA activities earlier into the development life cycle, ultimately achieving the desired result of QA implementation across the entire development organization.

Assigning QA Management Responsibility to Non-QA Professionals

Developing and implementing a QA strategy is a challenging task that should not be delegated to an untrained resource. Too often, organizations assign QA management responsibility to employees with limited experience. QA is a career path requiring many years of experience and firsthand knowledge in best practices and techniques.

Assigning someone with little experience in QA to head up a QA implementation project makes little sense; it is similar to asking a member of customer service to establish and manage a data warehouse. One does not want to assign this responsibility to someone whose primary job is in another area, leaving him or her with very little time to dedicate to the QA implementation effort.

The benefits of having a person available with extensive experience in developing and implementing QA programs cannot be overstated. QA professionals see problems across a variety of organizations. Differences in organizational size, culture, and politics will have a strong bearing on the QA processes that best fit an organization. An individual with practical experience will be better suited to identify which processes will work now, which can be implemented later, and which may never take hold.

Implementing Sweeping Changes

Changing too many processes at once is a prescription for failure, as human nature tends to resist change. Remember, when developing a QA strategy, it is important to focus on a few recommended changes for early implementation. Organizations are better served by initially making smaller modifications that have a higher probability of success. Once the group experiences several small wins, more people will embrace the larger changes planned for the future.

Most experienced QA professionals will recommend that QA processes be implemented in a phased approach. Allowing the organization to gradually implement, accept, and internalize changes will increase the probability for success of future changes. It also gives the organization time to evaluate the changes to ensure the new processes are having a positive effect on the development of applications. This may be recognized as a type of design of experiments (DOE) approach, popularized by Six Sigma. While DOE is a very methodical process of statistical analysis, it is not necessary to follow such a strict methodical approach. Once the organization has achieved a certain level of success, such as meeting the goals and objectives of the original QA strategy, it may be necessary to incorporate more stringent methods of statistical analysis to achieve even greater quality improvements.

Lack of Measurement

It is often said, "You cannot improve what you cannot measure." Implementing methods and processes of measurement are an integral part of every successful QA process improvement initiative. As discussed earlier, it is important to have a method of measuring the achievement of QA strategy goals and objectives. Testing also serves as a means to measure the success of other QA activities performed earlier in the SDLC. In addition, performing a cost/benefit analysis or statistical analysis will help an organization determine which QA processes should be implemented or continued.

All successful QA improvement programs must have a way to measure program effectiveness. No executive manager would (or should) continue investing money into such a program without some means to objectively measure the value and success of the new processes. For that reason, before implementing QA processes into an organization, one should first define and establish some key metrics to measure the success of the implementation. Some examples of metrics to track include: percent bad fixes, percent rework, cost performance index, cost variance, defect density, first run failures, schedule performance index, and schedule variance.

Clearly defining how these metrics will be tracked and used early in the initiative can establish an objective method for determining the success of the implemented processes. In other words, positive trends are expected in these metrics across multiple projects where various QA processes have been implemented. Metrics provide justification of the costs and allow organizations to continue to progress toward improved application quality.

Charting the Path

Implementing QA processes is a daunting task, especially if one proceeds without first:

- Developing and understanding the QA strategy
- Distinguishing between QA and testing
- Assigning qualified and experienced QA resources to manage the implementation
- Breaking down the implementation into a phased approach
- Establishing metrics that can objectively prove the benefits of the processes being implemented

Understanding these five common pitfalls will allow software professionals to steer their company's QA department in a positive direction, ultimately ensuring greater quality and success for the company and its applications.