

Beyond Testing

Adopting SQA Practices for Defect Prevention

New approaches for sustainable software quality improvements



EXECUTIVE SUMMARY

In an intensely competitive environment, mere testing will not do. Software vendors need to adopt fully integrated software quality assurance practices for continuous improvement throughout the development lifecycle. Yet often offshore development and testing environments only focus on late-cycle error identification. This paper addresses SQA methods that reduce the number of defects entering the testing phase and the eight characteristics of a high value quality assurance testing partner.



TABLE OF CONTENTS

Beyond Testing	4
SQA Practices for Defect Prevention	4
Introduction	4
Why Software Quality Assurance?	4
Intangible Benefits	5
The Role of the Software Quality Analyst	5
SQA Analyst Deployment	6
Barriers to Implementation	6
Ensuring Success	6
Conclusion	7
For More Information on Defect Prevention	7
ABOUT KEANE	8

TABLE OF FIGURES

Figure 1 Including SQA in quality initiatives reduces defect insertion rates	4
Figure 2 Orchestrating SQA Analyst Deployment	5



Beyond Testing

SQA Practices for Defect Prevention

Adopting software quality assurance (SQA) practices can be a difficult and painful process for an organization. While many companies have well-established SQA groups, they often work independently from development teams. Moreover, development team members often perceive SQA as inserting process for process' sake and yet another project schedule hurdle to overcome. Despite evidence that quality assurance yields positive returns, companies seldom adopt these practices easily.

Organizations can inject defect-prevention activities into the software requirements and development processes through a software quality analyst, whose role it is to define, institutionalize, and audit an organization's SQA practices. Rotating analysts among development teams — rather than keeping them wholly separate from this group — can ensure the consistent adherence to quality initiatives throughout an organization's software development efforts.

Introduction

LA comprehensive software quality initiative is twofold, involving software quality assurance (SQA) and software quality control. However, the difference between the two is more than semantics. Quality control is an activity that ensures that the product or service meets the company's defined quality standards. It is commonly known as "testing." Quality assurance, on the other hand, is the overall set of processes that an organization uses to achieve its determined level of quality. While the quality control effort consists of testing activities and concentrates on "defect detection," the quality assurance effort involves the entire software development lifecycle (SDLC) and focuses on "defect prevention."

Although an effective software quality initiative is made up of detection and prevention practices, organizations often implement only half of the quality program equation when they set out to improve their software products. Typically, companies focus on increasing the efficiency and effectiveness of software testing. For example, a company might define and execute testing-related activities, change test management techniques to be more reactive to business needs, or infuse the enterprise with a holistic approach to testing and defect detection. While all of these techniques are effective in improving testing for defects that were injected into an earlier phase of the SDLC, testing alone cannot stop defects from happening. That is where SQA techniques have a critical role.

According to the Institute of Electrical and Electronics Engineers, Inc., "once a piece of software makes it into the field, the cost of fixing an error can be 100 times as high as it would have been during the development stage."¹ An IT organization can increase its efficiency at preventing defects by introducing

and maintaining SQA practices. As an IT organization implements these activities, fewer defects enter the testing phase of the SDLC. Therefore, the quality and cost of software releases improves, and software testing gains the greatest possible return (Figure 1).

Why Software Quality Assurance?

Overall software quality is a function of how well the software is built, not how well it is tested. To improve the quality of its software products, an organization must attack the problem at its core when defining requirements and developing code. Techniques such as peer reviews, causal analysis, metrics management, traceability, software quality gates, and knowledge management repositories help to streamline the requirements and development process, reduce the number of defects, and improve the return on investment of the SDLC.

While there will be an initial upfront expense to establish proper quality assurance practices, the long-term returns will often justify the investment. By using defect-detection improvements in tandem with defect prevention, the overall quality assurance and testing organization can help the business benefit through software releases that enable the organization to focus on achieving its mission instead of issuing tactical, bug-fix releases.

Through the creative use of software quality analysts, an organization can

"According to IEEE, "once a piece of software makes it into the field, the cost of fixing an error can be 100 times as high as it would have been during the development stage."

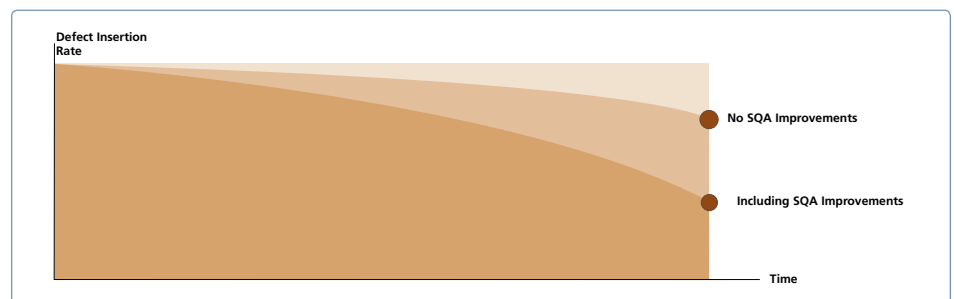


Figure 1 Including SQA in quality initiatives reduces defect insertion rates
Graphic for illustrative purposes only



adopt a method to institutionalize quality assurance initiatives. By rotating analysts among development teams firms can apply SQA initiatives consistently across an IT organization.

Intangible Benefits

There are also intangible benefits to be gained from rotating SQA analysts among development teams. Often the quality assurance organization is perceived as out of touch with the demands of the development teams and has difficulty implementing process improvements. There tends to be less resistance, however, if the quality organization embeds analysts in the development teams, reports to resident team management, and is accountable for the team’s success. Familiarity can lead to a cooperative environment. By experiencing each other’s pressures and demands, the quality analyst and the development team can create a

collaborative environment.

The Role of the Software Quality Analyst

The demand for outsourced testing is causing a proliferation of onshore and offshore testing service offerings. While these offerings share superficial similarities, they differ significantly in the business value they deliver. Gaining high-leverage, long-term business improvement takes more than a provider with a pool of available testing resources. No matter the partner, they will only bring marginal benefit if those resources are devoted to high-cost, late-cycle error identification. To gain sustainable improvements in quality, time-to-market, and change responsiveness, consider the eight characteristics of a high-value testing partner The analyst must work with project managers and team members to determine what the project manager considers achievable

and what the team members consider palatable. Attempting to turn an IT organization around and overcome all barriers to change is often not a realistic expectation; therefore, the software quality analyst must determine where the organization can realize the most benefit and how he or she can best use his or her time

Stakeholders must view the quality analyst as an enabler of best practices and an advocate for the requirements and development teams, not as a resistor who attempts to slow project progress.

The quality analyst must fully develop and document the selected quality practices and process outputs to the fullest extent possible and train all participants in the defined processes. The analyst is also responsible for auditing the organization on its compliance with the process.

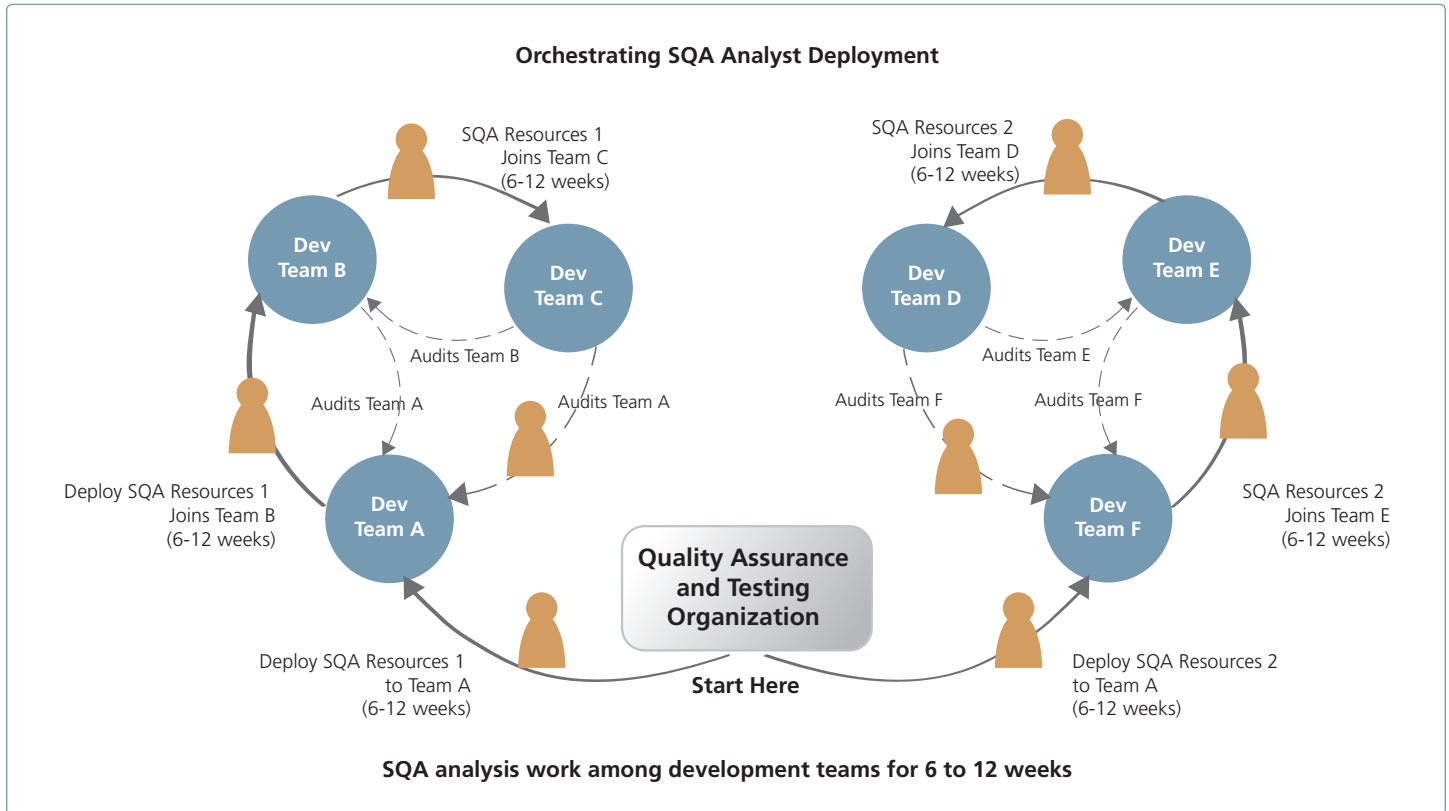


Figure 2 Orchestrating SQA Analyst Deployment



SQA Analyst Deployment

The proposed method is to deploy the SQA resources to development teams at the outset of a software development project. The quality analysts work directly with the development teams and participate in all their team activities as a member of the group.

By rotating quality assurance analysts among teams, IT organizations achieve several benefits: (1) they make efficient use of limited quality analyst resources, (2) they encourage development teams to adopt quality initiatives, and (3) they establish a routine of auditing development teams to ensure they have institutionalized the best practices.

The strategy for injecting quality into the software requirements and development processes is executed in the following major phases:

1. The analyst is deployed to a development team for 6 to 12 weeks. (This is a suggested timeframe. It allows enough time for the analyst to be incorporated into the team and to conduct the quality initiative.)
2. The analyst conducts a gap analysis against existing processes to determine the most appropriate quality practices to institutionalize.
3. The analyst conducts adoption activities, such as documentation and training.
4. The analyst conducts reporting and auditing activities after the quality practice is conducted.
5. The analyst is deployed to another team, auditing the previous team according to a pre-defined regular schedule.

The QA team management can determine the approach to conducting ongoing audits in terms of frequency and scope. Implementing this method is limited only by the number of quality analysts that a QA organization deploys and the degree to which it institutionalizes the process. QA organizations can implement audit teams

on a large or a small scale. As the number of requirements and development teams involved in the quality initiatives increases, organizations can develop additional audit teams.

Quality assurance analysts are always centrally managed by the SQA and testing organization. However, while in the deployment rotation, the quality analyst has two reporting chains. First, there is a dotted-line reporting chain to the resident requirements or development team project manager. In this relationship, the quality analyst reports on the status of the quality adoption project. Risks, issues, and hurdles are discussed, as well as the schedule and scope of the quality practice. It is imperative that the resident project manager is kept up-to-date on the status of the initiative. Additionally, the resident project manager is empowered to assign work to the SQA analyst resource within the overall scope of the quality analyst's mission.

Second, the quality analyst will continue to have a hard-line reporting responsibility to the quality assurance and testing director. This reporting relationship ensures a consistent application of processes across the organization. This reporting link is critical to adopting consistent, repeatable processes throughout all teams. Additionally, this relationship will serve as an outlet for status reporting and issue escalation. If, for example, the team resists adopting certain quality processes and the resident project manager cannot resolve the issue, the analyst can then escalate the issue to the quality assurance and testing organization director.

Status and progress reporting is shared in this relationship, as are risks and issues. This overall approach is owned by the quality assurance and testing group and, therefore, all strategic improvements in the process must be managed by that team.

Barriers to Implementation

Like any other initiative, there can be barriers to successfully implementing this model. Most notably, the project teams' dedication to quality assurance initiatives can be the most significant hurdle. The most common remedy is executive-level commitment. However, there are other techniques that the quality analyst can use to help increase the chances of adoption. For example, by working with the resident project manager to define the scope and schedule, the project team will more likely be able to include quality initiatives into their schedule without disrupting deadlines.

Other barriers include the resident project manager's lack of support and the team's inability to see and understand the quality initiative vision. Quality analysts must win management's dedication to the initiatives by illustrating, for example, costs saved over the long term by adhering to quality gates, conducting peer reviews, and developing a metrics management plan.

Ensuring Success

The most important characteristics of the quality analyst in this model are flexibility, understanding, and dedication. It is important that the quality analyst understand the team's pressures and then address them effectively. The analyst must remember that he or she is working with the team, not parallel to the team. Moreover, the quality analyst must adopt a "progress over perfection" attitude in order to influence the project teams and obtain success.

To increase the odds of success, look for senior quality assurance analysts who have the following skills:

- 10+ years of experience in SQA
- Familiarity with quality assurance models
- Good communications skills
- Project management experience
- Strong writing skills



- Understanding of SDLC models
- Understanding of software engineering process evaluation and auditing
- Previous success in designing, implementing, and measuring new process and quality initiatives, from conception to institutionalization

Often, firms choose to bring in an external partner capable of driving change in its QA programs. Third parties can prevent conflicts of interests and view processes objectively. Consider the following when selecting a quality assurance partner:

1. Business orientation to quality assurance

The ideal partner must understand the strategic business reasons for SQA practices and target the areas of greatest business importance.

2. Vision of improvement beyond finding defects

By focusing SQA practices on requirements definition and validation of specifications, your SQA partner will help you roll out completed systems that more closely meet business objectives.

3. Production experience

Real-world experience in supporting production applications is essential for understanding where and why to apply SQA initiatives to prevent

defects from impacting service and business performance.

4. Commitment to results

A trusted partner is willing to be held accountable for its performance through metrics and service level agreements. This commitment includes the quality of its results, the productivity of its teams, and the ability to leverage knowledge gained during SQA to continually improve development processes.

5. Project management expertise

A properly executed SQA program involves considerable coordination and communication to keep business objectives in sight while ensuring the flawless execution of countless details. A prospective partner's project management expertise is as important as its technical testing knowledge for guaranteeing the success of its delivered services.

6. Dedicated SQA and testing practice

To effectively deliver its services, an SQA partner must demonstrate it has the experience and resources to apply its SQA methodology to any environment.

Conclusion

Choosing to adopt software quality assurance best practices is a decision that organizations should make with

deliberation. Because it changes the way people work, institutionalizing SQA processes can create an environment of resistance. The method proposed in this paper enables the gentle adoption of repeatable quality processes that will ultimately improve software quality levels and IT's contributions to the business mission. In turn, operational units will be more satisfied with their IT counterparts.

Sources

1. Charette, Robert N. "Why Software Fails." *The Institute of Electrical and Electronics Engineers, Inc. (IEEE)* September 2005

For More Information on Defect Prevention

Making testing integral to your development process can improve the quality of your software. Request Keane's white paper "An Ounce of Prevention: The Importance of Testing" at http://www.keane.com/keane_ideas/whitepapers/1174_1208.htm.

Learn how to bridge the developer-tester divide to boost the quality of software development projects. Request Keane's white paper "Meet the New Tester: Laying the Foundation for Preventative Testing Practices" at http://www.keane.com/keane_ideas/whitepapers/1174_1207.htm.

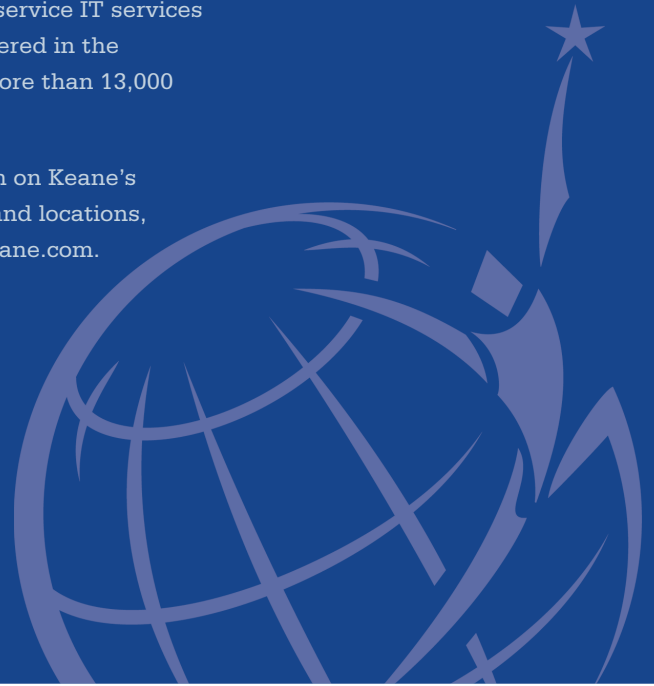


About Keane

Keane partners with businesses and government agencies to *optimize* IT investments by delivering exceptional operation, maintenance, and evolution of mission-critical systems and business processes. Keane helps clients realize the greatest value from their IT investments by leveraging an insider's hands-on understanding of the nuances and subtleties of their applications, processes and infrastructure making the recommendations we give more actionable, the work we do more pragmatic, and the results realized more measurable.

In business since 1965, Keane is an agile, mid-sized, full service IT services firm with headquarters in the United States and more than 13,000 employees globally.

For more information on Keane's services, solutions, and locations, please visit www.keane.com.



Keane

Corporate Headquarters
88 Kearny Street, Suite 1650
San Francisco, CA 94108

For more information about Keane's services, contact us at:

877.88.KEANE
info@keane.com
keane.com

