

Collaboration as an Imperative of Application Quality Improvement

ZELJKO PANIAN

The Graduate School for Economics and Business

University of Zagreb

J.F. Kennedy Sq. 6

CROATIA

<http://www.informatika.efzg.hr/zpanian>

Abstract: - Application quality is a team effort, and requires a complex combination of IT skills and roles. Nobody reasonable would dispute this notion, yet surprisingly, all too many IT organizations attempt to implement and assure quality in isolation. Ensuring the appropriate level of collaboration is a key IT management challenge. Software productivity tools and well-defined processes can help create an environment that facilitates collaboration. The paper examines desirable features of the solution supporting collaborative environment for quality.

Key-Words: - Collaboration, collaborative environment, application, quality, information technology, software productivity, integration, management

1 Introduction

When it comes to improving application quality, there is only so much an individual can do. Sure, each person can control the quality of his or her own work, but in Information Technology that work is invariably part of a much larger endeavor.

Placing a high-quality application in production is the culmination of a considerable group effort that includes a dynamic mixture of business and IT personnel analyzing, designing, architecting, programming and testing the components that compose the finished product. And that product is only as good as its weakest link. First-class programming cannot overcome a flawed business design, nor can superior testing cure every defect introduced throughout the development or maintenance process [1].

Moreover, almost all projects face budget, resource and/or schedule constraints that force tradeoffs and compromises throughout the project life cycle. Inevitably, these pressures affect quality. In some cases, making a quality tradeoff is a business decision, but in others, a compromise made in one area may unintentionally handicap the quality efforts in another. Even with the best intentions, the myriad people working on an application project are likely to differ on what constitutes quality in the first place.

For these reasons, achieving true application quality is necessarily a team effort. It requires common goals to ensure that everyone working on the project shares the same priorities and objectives. It needs standard roles and delineated responsibilities to avoid conflicts and ensure coverage for all necessary activities. It demands free exchange of information to shape efforts and

uncover conflicts and dependencies. Throughout the project, team members must continuously share, discuss, agree upon and prioritize actions based on rapidly changing information from many sources.

To achieve high quality, the entire team needs support. Ad-hoc approaches will never provide consistent results. First, quality efforts must integrate with development efforts and permeate the entire application life cycle, starting during planning and requirements gathering and continuing to final sign-off and production delivery. Second, team members must have well-defined processes and standards to focus and coordinate their efforts, ensure consistency and facilitate knowledge exchange. And third, they need tools to encourage communication, capture quality data, and automate otherwise complex and tedious testing and quality assurance activities.

In short, quality requires collaboration, along with an environment that encourages and supports it [2]. As the buzzword of the moment, the term "collaboration" is applied to all sorts of desktop, workflow and Internet solutions that are supposed to facilitate team efforts. Unfortunately, many of these solutions are too broad in scope and fail to live up to their hype. In contrast, Compuware focuses on a much more targeted form of collaboration. We define collaboration as a joint effort toward the common goal of delivering higher business value through enhancement of application quality throughout the application life cycle. Our vision of collaboration encompasses an environment, supported by technology, methodology and expertise, that builds and supports effective working relationships between all stakeholders in the application development process.

2 Higher Level of Collaboration Results in Better Quality

Nobody reasonable would dispute the notion that achieving application quality is a team effort, yet surprisingly, all too many IT organizations attempt to implement and assure quality in isolation. In these organizations, quality is often an afterthought, implemented by a separate QA team upon the completion of development activities. Unfortunately, this approach is not conducive to either quality or collaboration.

Many factors serve to inhibit effective collaboration and thereby reduce the effectiveness of quality assurance efforts and compromise the overall quality of delivered applications. Some of the most common factors of this kind include [3]:

- *Lack of development process integration* – Frequently, application quality assurance efforts are not integrated with the development process and life cycle. Relegated to the final phase before an application reaches production, quality assurance efforts are limited to the costly practice of catching rather than preventing defects. Schedule-wise, this positioning squeezes testing and assurance efforts, forcing them to fit between development overruns and fixed delivery dates, reducing their scope and further eroding their effectiveness.
- *Organizational silos* – The builders, testers, operators and receivers of application quality reside in separate organizations reporting to different managers. These organizational boundaries make change slow and difficult, and leave quality assurance (QA) teams with little influence over the practices responsible for creating quality in the first place. Worse yet, these silos isolate both the QA and development teams from the business organizations that define the original requirements, accept the results of development and depend on the applications in production.
- *Little commonality between organizations* – Organizational isolation means business, development and QA teams have their own distinct, and not necessarily compatible, processes, tools, priorities and visions for moving forward. Lack of a common language between organizations inhibits effective communications and results in piecemeal rather than end-to-end views of the decisions, activities and deliverables that comprise a development effort.
- *Poor information exchange* – Ineffective communications across organizational boundaries mean the QA team is often not aware of shifts in

priorities or changes that impact testing efforts, leading to ineffective targeting of efforts and other inefficiencies. Similarly, the development team is unaware of the impact of its decisions and rarely receives the feedback it needs to improve its efforts. While well aware of the budget and schedule, management has little visibility into the details of the project, its quality measures and the risk trade-offs that are made along the way. When cost and time-to-market pressures trump quality efforts, additional business risk is accepted without full awareness of its consequences, magnitude and probability of occurring

- *Operational inefficiencies* – Operational inefficiencies abound as a result of the forgoing issues. Work efforts are sequential, wasting opportunities to compress schedules through parallel efforts. There is little reuse of project and QA assets, resulting in significant duplication of efforts. Less efficiency means more defects reach production with significant downstream impact in terms of error and correction costs, business disruptions and compromised application value.

Although significant, these issues are not insurmountable. They cannot be addressed, however, solely through the QA team, nor can they be solved through reorganizations within IT. Organizational boundaries are a fact of life and the issues they bring will be exacerbated as IT moves increasingly to distributed, multi-sourced and multi-vendor organizational structures. The only solution is to establish an environment that overcomes these boundaries and encourages teamwork, communication and alignment across diverse groups. Collaboration is an imperative to enhance quality.

3 Building a Collaborative Environment for Quality

The need for effective collaboration as a means to improve quality is indisputable, but the tough question is how to achieve it. This challenge seems even more daunting for traditional quality organizations that face some or all of the issues described in the previous section. Management directives won't guarantee better teamwork, but creating an environment to enable and nurture collaboration will [4].

IT organizations can create this environment by following six key steps:

- a) Integrate business units into the quality process.
- b) Create an application quality life cycle.
- c) Govern through shared information.
- d) Externalize testing and development assets.
- e) Enabling virtual resourcing.

- f) Supporting collaboration with tools and processes.

Let us discuss these steps in some more details.

3.1 Integration into the Quality Process

As the ultimate recipients of application quality, business units are important stakeholders and information sources that must be part of the quality process. Quality starts with business requirements, and collaboration between QA teams, developers and business representatives during requirements gathering and planning is essential to develop a common understanding of business risks, priorities and expectations.

Requirements express specific business needs and can be valued and prioritized by cost, risk and benefit potential. Business risk analysis enabled by this information should drive all QA decisions. It provides a common language for discussing quality issues and trade-offs with business personnel and provides a business-oriented context for making decisions. Expressing QA issues in business terms keeps business areas interested and informed throughout the entire development cycle.

3.2 Creating an Application Quality Life Cycle

Integrating quality throughout the development process is made possible by creating an Application Quality Life Cycle that provides assistance and review at each stage of development.

The value of this integration over typical back-end quality approaches is enormous. As shown in Figure 1, catching a defect in production is as much as 100 times more costly than identifying and removing that defect during requirements definition.

Continuing quality efforts through each stage provide multiple opportunities to trap errors before they reach production. It helps to identify and resolve technical risks early, thereby minimizing reworks and better directing QA efforts. Encouraging teamwork between business representatives, developers and QA personnel at each stage breaks down organizational silos and allows these groups to work synergistically to achieve common quality goals.

3.3 Governing through Shared Information

As projects progress, change is inevitable and coordination and continuous alignment to common goals become crucial.

Information sharing between stakeholders and across organizational boundaries enables solid governance and provides the means to keep projects on track. Tracking progress and project status through roll-ups across all sub-projects, not just the projects within a given organization, is needed to ensure a complete view of the entire effort. Managing requests must take place through a shared tool that allows team members to share and automatically assign requests along with their associated tasks, enhancements, defects and changes. By working with these requests, business unit representatives can assign priorities and weigh requirements as they balance quality, schedule and cost constraints.

This shared information ensures informed go/no go decisions at each stage of the process.

3.4 Externalizing Testing and Development Assets

Development efforts create many documents, tools, test cases and other assets that can be shared. All too often, these assets remain hidden within the group that initially created them, leading to duplicate work or suboptimal execution within other groups. Externalizing these assets into a common area shared by all groups provides easy access and encourages the use of these valuable aids.

Sharing and reviewing documents, requirements, test cases and scripts between stakeholders add to the quality of these assets and facilitate reuse during each development phase and even across projects. Deploying a common quality methodology across organizations aligns and improves the efficiency of work efforts through shared processes and by encouraging optimal tool usage. Shared metrics enhance information exchange and enable continuous process improvement of development and quality efforts over time [6].

3.5 Enabling Virtual Resourcing

All development efforts have ebbs and flows in resource requirements. Adding resources to handle peaks in

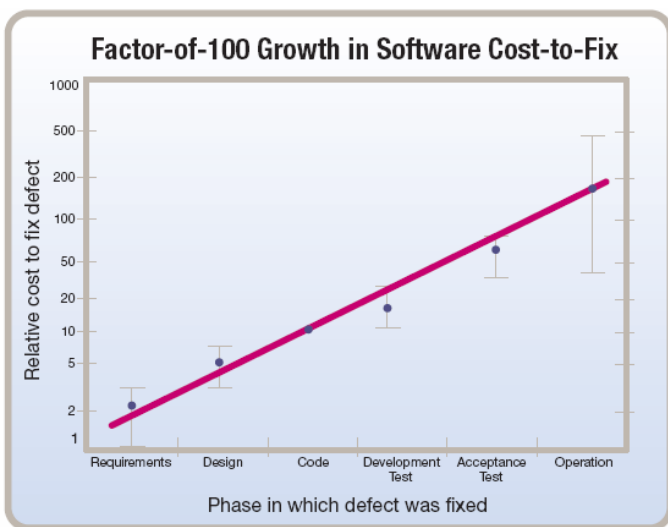


Figure 1 – Cost of Quality by Phase [5]

demand or specialized requirements is difficult to accomplish quickly in organizations that lack common tools and processes.

This difficulty increases when including external organizations, whether they are on-, near-, or off-shore.

Building a seamless community across groups requires greater process maturity and tool support to enable effective collaboration. Centralizing project management as a common service within the collaborative environment enables enterprise-level management of projects and resources regardless of the organizational distribution of project team members.

Providing web access to quality information and testing assets enables team members in any location to work together in a common environment and allows managers to govern all project resources based on the same types of shared information [7]. By supporting the productive use of external resources, organizations can assure the quality of application deliverables while being able to scale to meet changing needs and development cycles.

3.6 Supporting Collaboration with Tools and Processes

Software tools and processes are the ultimate enablers of application quality collaboration. They provide the consistency, productivity enhancements and controls that underlie the collaborative environment [8].

Central repositories enable the externalization and control of testing and development assets. Workflow automation facilitates communication, increases the efficiency of task and defect correction assignments and ensures compliance with common processes. Testing and tracking tools automate activities that would be too tedious, error-prone or unproductive to perform manually.

These tools not only improve efficiency, but also provide a source of data for metrics collection and analysis. Software metrics are essential not only for managing the development effort, but also to enable continuous improvement efforts throughout the application quality life cycle.

4 Desirable Features of the Solution Supporting Collaborative Environment for Quality

The appropriate software solution can enhance application quality through collaboration in several ways:

- *Enabling cross-organizational collaboration and virtual resourcing* – To enable collaboration, the appropriate software solution should provide a

broad environment that is deployed and accessed across geographic and organizational boundaries. It should include a web-enabled interactive workspace where developers and QA specialists could draw upon a breadth of available tools and shared test assets. These capabilities should allow dispersed teams to work from a common set of requirements, test plans and test cases to plan and execute the QA function. Manual and automated scripts should be sharable and work assignments changed. Full defect traceability should allow developers to review and fix defects quickly. Request management and workflow automation should help team members to share information and automatically assign and route requests

- *Gaining business-area involvement* – Through a combination of requirements management and focus on business risk analysis, the usable collaborative solution should provide the tools and common language to integrate business area personnel into collaborative planning and management efforts. Requirements management should ensure that testing efforts are business-driven and support the automatic creation of test plans and assets from requirements. Synchronizing test results with requirements should provide a business-oriented view of quality. The solution has to apply a variety of risk factors such as prior testing, function maturity, function complexity and past defects as part of a business risk analysis to optimize testing. Business users should be able to understand the impact of changing risks, cycle attributes and requirements through “what if” scenarios that let them balance quality, schedule and cost to meet their objectives.
- *Strengthening project governance* – The appropriate solution should centralize project management as a common service to enable enterprise-level visibility and oversight over multiple concurrent projects and teams distributed across multiple locations. It should provide real-time metrics and comprehensive standard and ad-hoc reporting of key quality and performance metrics. Alerts driven from standard thresholds should enable rapid response to change. Middle managers have to have visibility into ongoing projects through role-based portals which include reporting on standard metrics and the ability to drill down into any aspect of the testing function. Senior managers should gain insight through an executive dashboard that captures and rolls up information from multiple projects to provide an end-to-end view of the overall effort.

- *Full application quality life-cycle and process support* – Rather than just a point collaboration tool, the solution would be a comprehensive system that covers the application quality life cycle. It should bring a full complement of enterprise-wide tools that automate critical quality assurance tasks, including software analysis, test case creation, test execution, test management,

defect tracking and performance measurement along with a repository and web access to support test asset externalization. It should deliver process and best practices through its integrated quality methodology.

Full Application Quality Life-cycle and Process Support System is schematically shown in Figure 2.

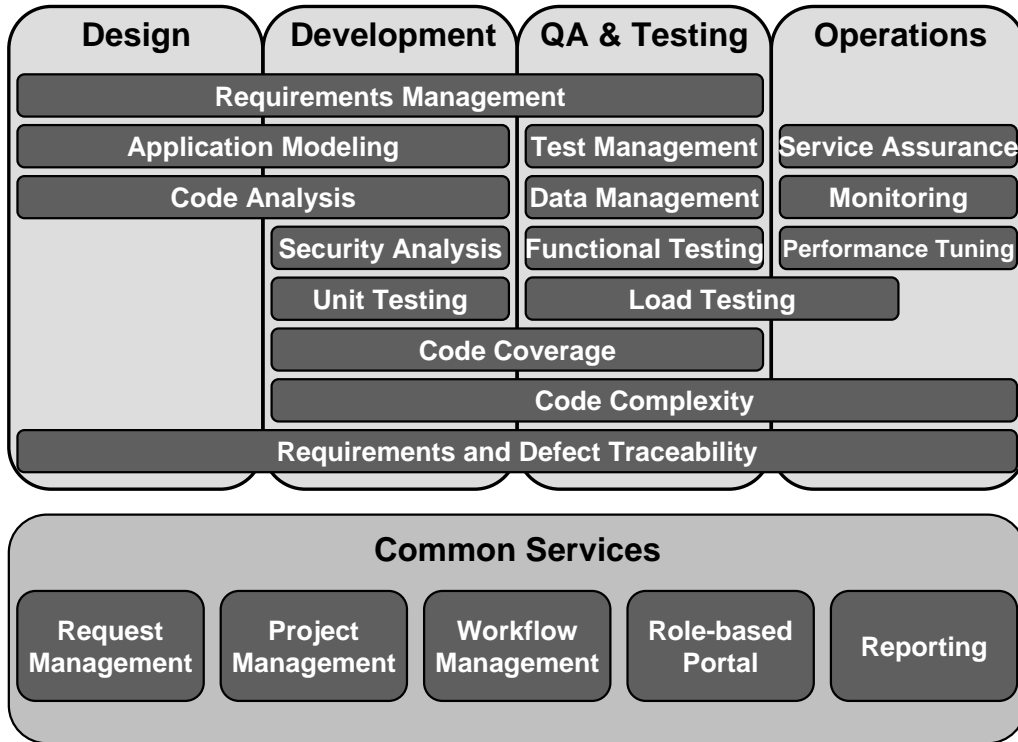


Figure 2 – Full Application Quality Life-cycle and Process Support System

5 The Benefits of Collaboration

Greater collaboration is much more than a means of encouraging better team work. Along with addressing its primary goal—better application quality—it delivers many additional benefits ranging from reduced business risk to resources freed for other strategic initiatives [9].

These benefits include:

- *Better project outcomes* – The bottom line is that the business gets better applications for its development, enhancement and maintenance investments. Fewer production defects means less disruption of business activities and closer fit to the application’s intended business purpose. Higher-quality applications cost less to operate, support and enhance and less money is spent correcting and recovering from errors. Reducing defects means lower risk of business, customer and financial impact from functional, operational and performance errors. In short, better

applications deliver higher value to the businesses they support [10].

- *Improved governance and communication* – The combination of better information exchange and integration of quality into an overall application quality life cycle breaks down artificial organizational barriers to enhance teamwork through stronger relationships and better communications between all stakeholders in the development process. Free exchange of information ensures continual alignment with project goals and better informed decisions and enables faster response to changes and issues. Sharing real-time project metrics ensures managers in all organizations are kept aware of project progress and any issues that need attention.
- *More efficient and effective requirements management* – Well-defined requirements drive

successful development efforts. Including the QA team in requirements definition ensures that all stakeholders share a common view of what needs to be delivered. Applying quality processes to requirements catches defects at a stage where they can be corrected with minimal rework, a far less costly option than catching those same errors in production. More importantly, these efforts ensure that the business gets what it wants; requirements that best match their needs and an ongoing plan to ensure that those requirements won't be missed during development.

- *More efficient and effective testing* – Collaboration improves all stages of testing, from planning through execution. It enables organizational synergies through shared plans, tasks and milestones that lower costs and reduce delivery timelines. It encourages the effective use of test automation at all stages of the application quality life cycle. It enables test-driven development approaches that improve the testability of deliveries and direct efforts to the areas most likely to harbor defects. It supports the identification and reuse of test assets to enhance test coverage and improve productivity. Most importantly, it allows the use of risk-based testing approaches to achieve the optimal balance of risk, cost and schedule. These approaches optimize test coverage around reducing business risk rather than focusing on technical outcomes.
- *Higher productivity* – Higher productivity means more work can be done with fewer people. It lowers costs and frees resources for other strategic and tactical assignments. Improvements in team communications mean more effectively directed efforts. More efficient testing reduces resource requirements and shortens the overall time needed for testing [11]. Higher quality results in less rework at all stages of application quality life cycle.

6 Conclusion

In conclusion, implementing quality improvement through collaboration makes strong sense in business and technical terms.

But it also makes sense from a people standpoint. Who wouldn't rather work in concert with their peers to achieve a common goal? The arrival of collaborative environments heralds the end of the perpetually squeezed, isolated QA organization and launches a new era of cooperation to deliver greater business value through higher quality.

References:

- [1] Fitzgerald, Michael, *Building B2B Applications with XML*, John Wiley & Sons, Inc., 2001
- [2] Ballou, Melinda-Carol, Best Practices for Application Quality Management: Improve Efficiency, Increase Business Relevance, and Cut Costs, <http://www.metagroup.com>, 06/2004.
- [3] Drakos, Nikos; Burton, Betsy; Cain, Matthew W.; Austin, Tom; Eid, Tom; Mann, Jeffrey; Smith, David Mario, Management Update: Predicts 2006: Collaboration Comes of Age, <http://www.gartner.com>, December 7, 2005
- [4] Silver, Bruce, Breaking New Ground with Collaboration, Content Awareness, <http://www.brsilver.com>, 03/2005
- [5] Basili, Victor; Boehm, Barry, CeBase Software Defect Reduction Top-10 List, Presentation, January 10, 2001, p. 5
- [6] Pivka, Marjan, *Kakovost v programskem inženirstvu*, Desk, Maribor (Slovenia), 1996
- [7] Panian, Željko, Distributed Information Technology Impact on Enterprise Change Management Projects, Keynote Paper, 2nd SENET Conference on Project Management", Cavtat (Croatia), 04/2002., Conference Proceedings on CD, str. 227-235
- [8] Malveau, Richard; Mowbray, Thomas J., *Software Architecture: Basic Training*, Prentice Hall PTR, 2004
- [9] Ozzie, Ray; O'Kelly, Peter, Communication, Collaboration & Technology: Back to the Future, <http://www.groovenetworks.com>, 2003
- [10] Morgenthal, J. P., Enterprise Architecture: The Holistic View: In a SOA Universe, How Important is Platform Selection?, *DM Review*, No. 7/2005
- [11] Boch, George, Object-oriented development, *IEEE Transactions on Software Engineering*, Vol. 12, No. 2, 1986, pp. 211-221