

AN ANALYSIS OF A SOFTWARE QUALITY ASSURANCE TOOL'S IMPLEMENTATION: A CASE STUDY

John J. Scarpino, Robert Morris University, scarpino@rmu.edu
Paul J. Kovacs, Robert Morris University, kovacs@rmu.edu

ABSTRACT

Software Quality Assurance has grown within the last 10 years and now incorporates more sophisticated tools that enhance software testing capabilities. However, before testing can be conducted properly, the correct procedure must be implemented. Too often, organizations rush into implementing a software quality assurance tool without first establishing a viable quality assurance process. This paper reports on data collected in December of 2007 concerning the implementation of a Software Quality Assurance tool at a Fortune 500 company in August 2006. The data analysis reveals problems that can arise while applying a Software Quality Assurance tool, the means required to employ a quality process and how to resolve major issues that may be caused by the initial implementation of the process.

Keywords: Software Quality Assurance, Software Quality Assurance Testing Tools, Software Quality Process Implementation, Software Quality Management, Software Test Automation, Software Development Lifecycle

INTRODUCTION

The fact that the development of all software systems should adhere to high quality standards is crucial in order to assure that errors and defects are identified and eliminated before the application is implemented. For a software product to achieve this quality, it must afford exceptional usability (the interface must let those who use the product accomplish their goals and tasks effectively and efficiently, while working in their own physical, social, and cultural environments [4]), functionality (a set of requirements or aspects linked with computer software), and compatibility (intended to act with another system or device without adjustment). Many organizations are therefore infusing Software Quality Assurance (SQA) throughout the Software Development Life Cycle (SDLC). This is quickly becoming an avenue through which “quality” testing is completed. The Handbook of SQA indicates that “SQA is the functional entity performing software quality assessment and measurement” [7]. SQA encompasses the complete SDLC, which includes processes such as software

design, coding, source code control, code reviews, configuration, and change and release management. SQA not only makes certain that an application is free of errors and defects, but that it is reliable, fully documented, maintainable, and completely functional.

Numerous organizations assume that with the purchase of a SQA tool, they validate process while eliminating any issues that are concurrent with implementation. Indeed, there are tools available that can assist SDLC project management by reducing the need for the “human touch”, at least to an extent. Such tools include: requirements-gathering for Business Analysts, test design tools for developers, and testing tools for Quality Assurance (QA) teams. But, in the article entitled *Quality Assurance: Much More Than Testing* Stuart Feldman states, “Quality Assurance isn’t just testing, or analysis, or wishful thinking. Although it can be boring, difficult, and tedious, QA is nonetheless essential.” [3]. Similarly, the IEEE Standard 12207 defines QA as “a process for providing adequate assurance that the software products and processes in the product life cycle conform to their specific requirements and adhere to their established plans” [3]. Therefore, quality is imperative and a company needs to make sure that it invests in both process and testing – not one or the other.

SQA consists of two major entities: process and testing. Process is the backbone upon which all of the functional aspects of SDLC rely. This is the entity that dictates when to do what, who should do it and how they should do it. Even if the process is expected to be informal, it still must be clearly mapped out before the SDLC is executed. Software Development procedural tools exist to make for a smooth implementation – but it is important to remember that the tool itself cannot create functionality of the product. Only real, live people can do this. People (testers in particular) are required to create a strategy, so process can exist in the first place.

SQA has grown within the last ten years and now incorporates more sophisticated tools that enhance software testing capabilities. This has allowed SQA testers to: find and report defects and issues faster;

make sure that their tests meet customer specifications and requirement-traceability; have a repository centralization for communication; archive the test plan and execute it manually or automatically; maintain reusable tests; and document, verify and audit which tests were run for each project.

Although software testing tools are adopted for use in many organizations, the way such tools are implemented often holds the key to success. Too often, organizations rush into implementing a SQA tool without first establishing a viable QA process. The SQA tool which was bought manages test requirements, test plans, test execution, defect management and software test automation. This paper reports on interview data collected in December of 2006 regarding the implementation of a SQA tool at a Fortune 500 company (**to be referred to as ABC**) in August 2006. The paper discusses problems that can arise while applying an SQA tool, the means required to employ a quality process, and how to resolve major issues that may be caused by the initial implementation of the process.

IMPLEMENTATION OF THE SOFTWARE QUALITY ASSURANCE TOOL

At the end of August 2006, ABC had completed the installation of a SQA tool. The SQA tool's use was supported by upper management, with the expectation that it would increase efficiency by replacing human testers with software test automation. The purpose of this SQA tool was to write requirements for test planning, manage test design and execute test plans, find and record issues/defects, as well as automate test functionality. Forty functional departments in ABC were divided among this new system. Not all departments were required to use the SQA tool, but many departments did.

Software test automation permits a program to run automated tests with as much (or more) efficiency as if it were being done manually, which helps save both time and money. Upper management at ABC felt passionately about the return on investment possibilities by automation and therefore had high expectations of the project's outcome. The "reduction of headcount" was one such expectation. Senior Management was convinced that the automation piece of the SQA tool was the definitive answer.

During the initial implementation of the SQA tool in August of 2006, ABC's QA Center of Excellence (QA COE) employed one in-house Senior Manager,

one in-house Project Manager (who was unfamiliar with the tool), one in-house Business Analyst (also unfamiliar with the tool), and two outside technical consultants who were deemed experts with the tool. By June 2007, the manager of the QA COE was dismissed; the Project Manager transferred to another department; and the two technical consultants left the organization. The next month, the Business Analyst became manager of the QA COE. Then in August 2007, another Business Analyst was hired, and in September 2007 the first in-house technical expert with the SQA tool and process implementation was hired. However, by December 2007, the Business Analyst hired in August was transferred to another department. As of March 2008, the current team consisted of one manager of the QA COE and the in-house QA expert for 37 departmental areas.

The COE employees were expected to know how to use the SQA tool after one, high-level introduction class given by the technical consultant who helped implement the tool when it was first introduced in August 2006. Additionally, one Web-based training tutorial was available for education purposes, and a monthly group phone conference discussion was conducted to evaluate progress. Very few specific SQA documents were available and even the documents that did exist were buried deep within the ABC's Intranet Web site. No handout or brief document was created as a quick reference guide. Many of those who were in charge of maintaining the SQA tool had no consistent interaction with the users of the system. Users were left uninformed as to how the application should operate and there was no true, systemic process put in place to teach others how the application was supposed to function. There were a few users who volunteered to act as a pilot group to test the system. These groups dedicated their time to use the system more frequently, but it was still not used to its full capacity.

In November of 2007, upper management wanted to know the status of the project. There was awareness that the SQA tool was not being enabled across the organization as effectively as it could. As a result, upper management decided to employ an external consultant to evaluate the project and make recommendations. Since the QA COE was within the Corporate Project Management Office, its manager also welcomed the external consultant. Additionally, the in-house QA expert, who was hired in September of 2007, decided to conduct a parallel investigation with the external consultant. The information gathered by the in-house QA expert serves as the basis for this paper.

DATA COLLECTION

The data for this paper was gathered from interviews of seven product-facing users who represented many of ABC's departments. Each two-hour interview was conducted one-on-one over a period of five days. The data collected through this study was compiled by means of open observational analysis and documentation – each time an interviewee mentioned an issue it was documented and assessed by how many times it was indicated during that individual interview. All of the interviewees were introduced to the tool during its initial implementation. As expected, these front-line individuals were able to provide better, more qualitative data than would, say, a Senior Executive that did not use the product very often or who was not “hands-on”. The interviews were conducted by the both external consultant and the in-house QA expert.

An outline of the topics of questions asked during the interview process is as follows: Business Process Optimization, Defect Management & Control, Governance, Knowledge Management, Metrics & Dashboards, Post Installation, Product Installation, QA Environment, Release Schedules, Review and Inspection, SDLC used, Test Audit, Test Automation

Techniques, Test Kickoff, Test Planning and Design, Test Requirements Review, Testing Tools, and Test Work flow.

Specific issues expressed by the interviewees were recorded and occasionally interviewees were drawn into further discussion after they answered, so as to enhance documentation. The in-house QA expert was present for seven of the ten interviews, while the manager of the QA COE was present for three. No information was formally collected and documented by the manager of the QA COE.

FINDINGS

In-house Evaluation

The data gathered by the in-house QA expert from the interviews conducted of the seven product-facing users was categorized into nine issue areas, as to how many times each interviewee indicated an adverse response. The nine issue areas are listed in Table 1.

These nine major issues were also categorized in the order of highest to lowest severity. This data count is provided in the Table 2.

Table 1. Nine Issue Areas

Major Issue	Description
Assistance and Training	A lack of ABC's support of the tool and little education for its users.
Automation	Testing should have been conducted with software testing scripts, which would run and execute any testing that should need to occur.
Initial Implementation & Communication	There was no educational promotion and communication of the tool prior to its implementation.
Process	A clear idea of how the new system would flow internally/externally within the Software Development Lifecycle was non-existent.
Resistance	Too many employees circumvented a smooth transition between systems.
Time	A severely small amount of time was allotted in which employees were forced to accept and begin the implementation of the new tool.
Tool Analysis & Verification	There was only a short user-review of the tool to assure that its functionality met the ABC's expectations, and that all of the elements of the application would be used to the fullest.
Tool Layout	The application's design and set-up; both internally with the Software QA system and externally, with other resources.
Governance & Management	The QA department and ABC at large should have had a consistent review of the project's status and how the tool was being used on a daily basis.

Table 2. Issue Areas Data Count

Product Facing User	1	2	3	4	5	6	7	Total	Percent
Interview Date (2007)	12/6	12/6	12/10	12/11	12/11/	12/12	12/14		
Process	6	7	1	4	1	5		24	28.92%
Assistance/ Training		7	2	1	2	1	2	15	18.07%
Tool Layout	6	2	1		1	2	1	13	15.66%
Initial Implementation & Comm.	3	1	1		1	1	2	9	10.84%
Time			1	2	3	2	1	9	10.84%
Tool Analysis & Verification	1					3	1	5	6.02%
Automation		2	1	1		1		5	6.02%
Resistance					1		1	2	2.41%
Governance /Management							1	1	1.20%
Total	16	19	7	8	9	15	9	83	100.00%

Below is specific information as well as several quotes from the interviewees as indicated during the interviews for each of the nine issue areas:

Process

A suitable software-implementation Process requires a fluid, corporate-wide structure and positive employee communication. It is possible that ABC’s employee turnover and lack of supervision caused this most crucial stage fail. Process is the key to any successful tool implementation, as the saying goes, “Process First, Testing Second”. Unfortunately, the idea of having a true QA department was a new practice at ABC. In fact, the majority of ABC’s business groups employed “business analysts” who were expected to take on a QA role in addition to their other duties. The physiological transition of job-description also may have caused new users to become confused, since it was the first time they had worked solely within a QA role.

Quotes from interviewees:

“There is no documentation to tie-in the project management tool and the SQA tool.” / No one “manages severity and changing of priorities.”

Assistance & Training

Without proper support of the new system and its implementation-procedure, or the ability to aide users when they need help, the tool will not be a success. Develop documents for training and individualized support - especially if it is all new to employees.

Quotes from interviewees:

“How do I create reports?” / “Training is needed.” / “I do not know how to attach a spreadsheet to the test.”

Tool Layout

Tool Layout is a major component of a software system’s design. Therefore, a new application cannot be used properly unless the user has a working knowledge of its design. Managers should be in communication with their employees to assure that the application is being handled properly, and that all expectations are met.

Quotes from interviewees:

“Link the tool from the Project Management tool to the SQA tool.” / “We need access to more then one project within the SQA tool.” / “Test scripts and requirements are not mapped.”

Initial Implementation & Communication

Initial Implementation & Communication was not given to many of the system’s users. Instead, they were expected to use the system on a whim, figuring it out as they went along.

Quotes from interviewees:

The “Implementation of the Project Management Tool and the SQA tool together confuses things up.” / “Both the PM tool and the SQA tool was implemented at the same time but not communicated together.” / “The functionality of the PM tool and SQA tool as one was not communicated.”

Time

Time constraint, as mentioned before, was always an issue that contributed the lack of support. The ability to smoothly move into the process is necessary in order for employees to accept the tool and its functionality.

Quotes from interviewees:

"They want us to implement the tool but no time is allotted for it." / "QA estimated is done by management verses development estimate is done by the developers." / "Inputting the information from requirements to the functional flow takes time." / "Converting all of our test plans from spreadsheets to the SQA system takes too much time." / "It takes too much time to learn the system. The project will be over by the time we finally do." / "QA has been pushed back to two days of testing. The overall schedule can't change at this point, so the use of the tool and actual QA will be cut." / "There is no time to create internal departmental standards for the usage of QA."

Tool Analysis & Verification

In order for users to accept the system, it must be analyzed and then verified to meet ABC's standards. If the SQA tool has important elements that the user remains unaware of, or if it holds the capacity to be customized but no one knows it, then there is a good chance that it will be used incorrectly and not meet expectations.

Governance & Management

Governance and Management was almost not even a topic during the interviews, but it did receive one mention. Users must be watched to ensure that project testing and the process through which a tool is used meets the targeted timeline and risk assessment. When a user indicates that Governance & Management is needed, they indicate that no ownership was made. This is bad sign. Creating statistics and assuring that the tool meets the customer expectations is a key for success. Without these types of measurements the tool will fail.

Automation

One of the biggest assets a SQA tool can bring to a company is the ability to create automation scripts. These scripts are used to run functional tests as if a human was manually running it himself. The automation of a system helps create ROI, due to its ability to test at a faster and reusable pace - thereby allowing software testers to test other functional areas or more detailed test plans. It is very important that the users have access and knowledge to create these scripts. The Senior Executive of ABC mentioned this as one of the factors for investing in all of these tools, and it must be implemented after the SQA framework.

Resistance

Areas where users refused to accept the change of having a new SQA tool was a problem throughout its implementation. In order for overall acceptance to

occur, the users must see some value in the change. When upper management says "go" and users say "no", it is important as a manager to understand the concerns of all the users so that they can be heard.

External Evaluation

The evaluation undertaken by ABC's external consultant from the ten individuals interviewed resulted in the following issues.

1. Organizational Process Improvement
2. Testing Process Improvement – the segregation of duties on testing activities
3. Tool Implementation Improvement – the need for communication to drive employee acceptance and establish automation strategy
4. Include both management and practitioners in the process improvement effort to increase ownership and employee acceptance
5. Establish governance structure for review and status reporting.
6. Build a detailed process and implementation plan with clear goals and objectives
7. Simplify the documentation of the implementation-process
8. Build employee competency on tools
9. Define the tool's deployment strategy and ABC's approach
10. Identify individuals who are experts on the SQA tool
11. Establish inter-departmental communication and a have consistent progress-reports

PROBLEM RESOLUTION

As a result of this investigation, there was the formation of a QA process within the Information Technology department at ABC. The areas that ABC wanted to improve upon were: building fundamental processes in place; training the employees; establishing a pilot program for few departments and assessing their status with metrics; and providing adequate support for all users. From these improvements, the implementation will grow on a larger scale to be company-wide.

A plan was initiated to construct a process for the second attempt of the SQA tool implementation. The implementation process was to be completed in increments within only four departments, utilizing one department at a time. The first four departments were to serve as a "pilot," an example to other departments, and should be completely integrated within six months. The company intends to weave this process throughout the enterprise by the start of

August 2008. As of March 2008, work on the foundation is being conducted and the information is being collected by the in-house QA expert. The Corporate Project Management Office also wants to have scripts developed to determine a positive return on investment as soon as possible.

The following are the high-level milestones for this new implementation, in ascending order:

1. Work on "Baseline"
2. Preparation for Pilot
3. Review and sign-off on "Baseline"
4. Training sessions
5. Hire 1 additional resource
6. Deliver training – for the SQA tool and automation tool
7. Analyze and Implement "Baseline" to the 4 pilot departments – which includes developing test plans and automating the departments software for testing
8. Governance – use metrics to assure the status
9. Completion of the QA process

CONCLUSION

The top three issues recorded from the interview process were 1) Process, 2) Assistance & Training, and 3) Tool Implementation Improvements. There was no educational promotion and communication of the tool prior to its implementation and lack of process, too, spiraled to become an issue, along with employee turnover. Additionally, somewhere in middle-management a power-struggle ensued between employee and corporation and resistance became a major roadblock.

Since the QA process shapes a Quality Analyst's behavior and mode of thinking, it is not always easy to allow new ideas to take their course. Having a true QA process was a new concept for ABC, especially since most of the testing was previously conducted by Business Analysts. In essence, the SQA tool was unfit for the kind of method ABC already had in place. ABC implemented the SQA tool without first determining the correct layout and how to support it. Departments were selected to start the pilot, but no plain objective was given. ABC had a low capacity for support, since there were only two individuals (consultants) who truly understood the inner-workings of the product when it was first introduced. In September 2007, the new manager of the QA COE (formerly the Business Analyst) decided that the tool should be implemented by a "grassroots" approach, or "word of mouth".

Upper Management of ABC recognized the importance of the software product and implemented it with the expectation that it could save time and money with automation – but this is not where the emphasis should have been. ABC should have focused on developing a sound process for the SQA tool framework, and then have the automation scripts work both within the process and the tool.

Many times the senior management responsible for implementing a SQA tool operate under the same mindset that was indicated in an article written by Michael Donat "manual testing is too tedious, expensive and inconsistent to be effective." [2] But Donat goes on to say, "while automating testing, [it is] found [to be] very labor intensive to maintain a set of scripts describing each machine's portion of a given test. Maintainability suffers because the test description is spread over several files." Those responsible for implementing a SQA tool are not aware that maintenance could become a problem due to requirement changes and that issues would most likely arise during the process. The delivery of requirements marks the start of changes for software development and testing. Most organizations segregate their Business Analysts (BA) from the Development and Testing teams. In this case study, the BA team was conducting the testing and also acted as the Quality Analyst. Issues can be easily overlooked if the team that creates the software's requirements is also testing the application.

Those organizations that have a corporate Project Management Office need to separate Project Management, QA, and Business Analysis as separate managed groups. The reason for such a separation is to ensure that no bias exists among separate software life cycle entities. In those organizations such as ABC that primarily operate under the Mainframe platform, many of its systems are asynchronous when testing. A Business Analyst can conduct testing and documentation – but, as Web based and GUI systems become more prevalent, a door is opened to the possibility of having more holes in the functionality of the system. This makes for the need to have a separate QA group at the Center of Excellence level of the organization and also at the departmental level. Groups that only have Business Analysts to conduct the testing and documentation are realizing that they must create a separate QA group. This is especially true for the groups that are using Graphical User Interface (GUI) or Web based environments. Business Analysis and QA are two separate practices and must be conducted as such.

The need to have qualified SQA personal has been a dilemma for many years. Universities and Colleges need to develop courses and programs to educate students in both QA process and its tools. It is difficult to find qualified individuals in a growing area where Software Quality is becoming more of a demand. Many Quality Analysts fall into this field after being a developer or Business Analyst for many years but this field should develop employees from inside the class room rather than from inside the workplace. There is a growing need for experts in SQA tools and QA process.

Finally, the framework or “backbone” which includes Process and Assistance & Training is fundamental to any product execution. Always remember: “Quality Process first; quality testing second.”[6].

REFERENCES

1. Covey, S. R. (1989). *The 7 Habits of Highly Effective People*. New York: Free Press.
2. Donat, M. (2006). *Orchestrating an Automated Test Lab*. Retrieved February 10, 2008, from Association for Computing Machinery: <http://www.acmqueue.org/modules.php?name=Content&pa=showpage&pid=276>
3. Feldman, S. (2006). *Quality Assurance: Much More than Testing*. Retrieved February 10, 2008, from Association for Computing Machinery: <http://www.acmqueue.org/modules.php?name=Content&pa=showpage&pid=276>
4. Hackos, J.T., & Redish, J.C. (1998). *User and Task Analysis for Interface Design*. New York: Wiley.
5. Murphy, N. (2002). *A Question of Quality*. *Embedded Systems Programming*, 15(2), pp. 30-35.
6. Scarpino, J. (2008). *Quality Process First; Quality Testing Second*. Retrieved March 8, 2008, from Software Quality News: http://searchsoftwarequality.techtarget.com/news/column/0,294698,sid92_gci1289241,00.html
7. Schulmeyer, G., & McManus, J.I. (1999). *Guide to Software Quality Assurance*. (pp.7) New Jersey: Prentice Hall PTR.
8. Ward-Dutton, N. (2007). *Governance Ensures Consistency Across Projects*. Retrieved February 12, 2008, from Computer Weekly: <http://www.computerweekly.com/Articles/2007/10/01/227089/governance-ensures-consistency-across-projects.htm>