A POST ANALYSIS OF A FORMALIZED 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 (SQA) is a systematic approach to the evaluation of the quality of and adherence to software product processes that are established and are followed throughout the software development life cycle (SDLC). Within the last decade, (SQA) has become a required functional process in SLDC and now incorporates more sophisticated tools that enhance software testing capabilities. However, organizations can sometimes rush into implementing a (SQA) tool without first establishing a viable (SQA) process. To assure that testing is conducted properly, the correct procedure must be implemented.

This paper will act as a follow-up to the research conducted in December 2007 regarding the informal implementation of a (SQA) tool, which occurred at a Fortune 500 company in August 2006. New research data was collected in December 2008, after the Software Quality Assurance tool was formally implemented at that same Fortune 500 company. This paper will investigate if the formalized implementation in 2008 had a greater impact on the users of the (SQA) tool than the informal implementation in 2006. Data analysis will reveal problems that can arise during Software Quality Assurance tool implementation, the means required to employ a quality process, and how the company could have overcome major challenges that were caused by the initial informal software implementation.

Keywords: Software Quality Assurance (SQA), Software Quality Assurance Tools, Software Quality Process Implementation, Software Quality Management, Software Test Automation (STA), Software Development Lifecycle (SDLC)

INTRODUCTION

Establishing standards and procedures for software development is essential in order to assure that errors and defects are identified and eliminated before the application is implemented. For a software product to achieve this level of quality, it must allow 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, therefore, are incorporating Software Quality Assurance (SQA) throughout the Software Development Life Cycle (SDLC). This is quickly becoming a means through which "quality" testing is completed. Schulmeyer and McManus (1999) indicate 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. Not only does SQA assure that an application is free of errors and defects, but that it is reliable, fully documented, maintainable, and completely functional.

Numerous organizations assume that by purchasing a SQA tool, they validate process while eliminating any issues that are concurrent with implementation. Certainly, there are available tools 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 *Quality Assurance: Much More Than Testing,* Stuart Feldman [3] indicates that Quality Assurance isn't just testing or analysis. Although the QA can be difficult, boring and tedious, it is essential. Therefore, quality is imperative and it is necessary for an organization to ensure 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 consider that

the tool itself cannot create functionality of the product. Only individuals can accomplish this. Testers in particular are required to create a strategy, so process can exist in the first place.

Within the last decade, (SQA) has become a required functional process in SLDC 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. This paper reports on interview data collected in December of 2008 regarding the formalized implementation of a SQA tool at a Fortune 500 company (**to be referred to as ABC**). Analysis of this interview data compares findings of the formalized implementation from 2008 to the findings from the informal implementation of 2007. This study investigate 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

In December of 2008, ABC had completed the installation of a SQA tool. The planning and documentation process for the implementation lasted from January 2008 to July of 2008. The implementation occurred from July 2008 to December of 2008. The use of the SQA tool 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. The formalized implementation was conducted within four departments: Capital Markets, Commercial Lending, Sales & Service, and Wire Transfer. The eCommerce and Consumer Loans

departments were indirectly affected, and were then included in the formalized implementation.

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 was enthusiastic about the return on investment (ROI) 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 formalized implementation of the SQA tool in July of 2008, ABC's QA Center of Excellence (QACOE) employed one in-house Senior Manager, one in-house expert who was part of the planning process from January 2008 to July 2008. In May, they hired a second in-house expert to assist with the implementation, starting in July. Though the formalized implementation was taking place, the QACOE still had the responsibility of overseeing 37 departmental areas for in-house quality management.

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

1. Work on "Baseline"
2. Preparation for Pilot
3. Review and sign-off on "Baseline"
4. Training sessions
5. Hire one additional in-house expert resource
6. Deliver training – for the SQA tool and automation tool
7. Analyze and Implement "Baseline" to the four 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

During the formalized implementation the QACOE conducted classes and one-on-one sessions with each of the departments to analyze where they currently were to assist them with gaining the tools benefits with further functional growth and knowledge of the tools. Several documentations were provided as a process guide and training materials: QA Primer, QA Tools Manual, Quick Reference Guides and Educational PowerPoint Slides. The QACOE was also available over the phone, through e-mail and instant message if a quick question or guide was needed by any of the four main departments or the

other two departments. The QACOE did not provide a person to be solely dedicated to the four departments, because they also had to oversee the other quality management departments at ABC. In addition, monthly group phone conference sessions were conducted to evaluate progress.

In January of 2008, upper management initialized the formalized implementation project or the planning phase. The actual implementation within the departments was completed in July of 2008. The intention was to have at least four departments fully functional and develop best-practices so that other departments at ABC could follow the same method.

DATA COLLECTION

The data for this paper was gathered in December of 2008 from two hour interviews of seven product-facing users who worked in the departments where the implementation took place. Two of the individuals that were interviewed for this study were also interviewed for the previous study conducted in December 2007.

The data collected was compiled by means of open observational analysis and documentation – each time an interviewee mentioned an issue, it was documented and measured according to how many times it was brought up during that particular 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 who did not use the product often or who was not "hands-on." The interviews were conducted by ABC's only in-house QA expert.

A list of topics from 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 Workflow.

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 the only person present for the interview besides each interviewee.

FINDINGS

The data gathered in December of 2007 by the in-house QA expert was categorized into nine overarching issues according to how many times each interviewee indicated an adverse response. The nine issues are listed in *Table 1,* and were also used as a baseline for the December 2008 interviews. Four additional issues arose during the 2008 interviews, and these are listed in *Table 2*. These four additional issues were broken down into thirteen major themes that were encountered during the software quality assurance tool's implementation in 2008. These thirteen themes were categorized in the order of highest-to-lowest severity, with the four new issues highlighted in bold text as shown in *Table 3*. The data count from the interviews conducted by the in-house QA expert of the seven product-facing users is also provided in *Table 3*.

The original nine issue areas from the 2007 study were also categorized in the order of highest-to-lowest severity. The data count from the interviews conducted by the in-house QA expert of the seven product-facing users is provided in *Table 4*.

Table 1.

Original Nine Issues

Assistance and Training	Lack of corporate support for the tool and little education available for 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 and 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 and 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 during which employees were forced to accept and begin the implementation of the new tool.
Tool Analysis and Verification	There was only a short user review of the tool to ensure that its functionality met company expectations, and that all of the elements of the application would be used to their fullest.
Tool Layout	The application's design and set-up, both internally with the Software QA system and externally, with other resources.
Governance and Management	The QA department and the company, overall, should have had a consistent review of the project's status and how the tool was being used on a daily basis.

Table 2.

Additional Four Issues

Resources	The lack of human presence within each department throughout the implementation.
Lack of Upper Management's Knowledge of Tool and the Tool's Benefits	Management's ignorance of how the software quality assurance tool is used and how its benefits are actually obtained.
Company Needs to Listen to Internal Experts Recommendations	The corporation's non-use of the in-house expert's knowledge and guidance throughout the implementation.
Organizational Layout	Issues with how departments are organized and managed throughout the implementation.

Table 3.

New 2008 Research Data Graph

Product Facing User	1	2	3	4	5	6	7	Total	Percent
Interview Date (2008)	12/18	12/18	12/23	12/23	1/12	1/13	1/15		
Time	4	4	4	4	8	2	5	31	21.38%
Process	4	1	5	5	4		2	21	14.48%
Assistance and Training	3	3	5	4		1		16	11.03%
Resources	3	1	4	4	1	1	2	16	11.03%
Initial Implementation and Communication	2		4	4	3			13	8.97%
Tool Layout	2	6	1		1		1	11	7.59%
Automation	1		1	3	2		2	9	6.21%
Tool Analysis and Verification	3		2	1	1			7	4.83%
Governance and Management	3		1	1	2			7	4.83%
Lack of Upper Management's Knowledge of Tools and the Tool's Benefits.			1		3		2	6	4.14%
Resistance			1	1	2		1	5	3.45%
Company Needs to Listen to In-House Expert's Recommendations			1	1				2	1.38%
Organizational Layout Issue					1			1	0.69%
Total	25	15	30	28	28	4	15	145	100.00%

Table 4.

Original 2007 Research Data Graph

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 and Communication	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 the information that was indicated during the 2008 interviews for each of the thirteen major themes. Seven of the top issue areas contain several examples of quotes from the interviewees.

Time

Time constraint, as previously mentioned, was always an issue that contributed the lack of support, ability to understand the product, ability to understand the process, the review existing and future changes, and the ability to retain new information and knowledge. The capability to smoothly move into the process is necessary in order for employees to accept the tool and its functionality. Time is needed to accurately implement the tool without any impending factors.

Quotes from interviewees:

- ⇨ "Management told us not to automate because of time."
- ⇨ "We were not able to audit our teams testing because of time."
- ⇨ "More time is needed for the implementation."
- ⇨ "If time was there to learn and to go to seminars we could be successful.
- ⇨ "Automation is time consuming to fix."
- ⇨ "Time is needed to setup new scripts."
- ⇨ "The Software Quality Assurance tool take a lot of setup time."
- ⇨ "I don't have the time to train myself."
- ⇨ "No review of tests because of the lack of time."
- ⇨ "Time was too fast for the implementation."
- ⇨ "Still need time for the Test Planning and Design."
- ⇨ "Test execution is not complicated but need time which makes it complicated."
- ⇨ "Needed dedicated time for upfront analysis."
- ⇨ "More devoted time from a full-time person."
- ⇨ "The implementation seemed an afterthought not too much time was expressed."
- ⇨ "We had no time to be able to use the software quality assurance tool to the fullest, only the defect tracking tool though we did look at the other pieces of the tools functionality."
- ⇨ "It too time getting use to the tool."

Process

A suitable software-implementation process requires a fluid, corporate-wide structure and positive employee communication. Accurate documentation, communication and follow-up of ABC's software quality assurance process and the software quality assurance tool's process are crucial for accurate implementation. Unfortunately, quality assurance 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 be the first time they had worked solely within a QA role.

Quotes from interviewees:

- ⇨ "Need standard reports."

- ⇨ "Did not review automated or manual tests because we are the subject matter experts."
- ⇨ "No process optimization."
- ⇨ "The process document that was sent out by the QACOE was too late and did not truly integrate the team."
- ⇨ "No considerations on teams current performance and process."
- ⇨ "No metrics and reports."
- ⇨ "What happens after automation?"
- ⇨ "No review of manual tests."
- ⇨ "No good understanding of the Software Development Life Cycle (SDLC) workflow."
- ⇨ "Need more input from Business Analysts (BA) and team effort, rather then BA seeing it as busy work."
- ⇨ "The line of business does not use the software quality assurance tool."
- ⇨ "The process documentation hit us hard."
- ⇨ "SDLC no help."

Assistance & Training

Without proper support, guidance and training 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.

Quotes from interviewees:

- ⇨ "Lack of understanding how to create scripts some still don't know how. Peer review help needed to ensure scripts are done correctly."
- ⇨ "Both software quality assurance tool and quality assurance training needed."
- ⇨ "Internal help is needed rather then contractors."
- ⇨ "More training."
- ⇨ "Knowledgeable quality assurance person needed in group to guide."
- ⇨ "Automation training was done too late."
- ⇨ "We don't know how to apply the software quality assurance tool."
- ⇨ "Need guidance on business analysis (BA)."
- ⇨ "Still need guidance to structure the tool."
- ⇨ "Need upfront guidance with the software quality assurance tool."
- ⇨ "Need to take advanced automation training."

Resources

Resources are needed within each of the four departments and within the QACOE for its support and implementation. Very few resources were available for support of the four departments and

within the QACOE. The implementation of the software quality assurance tool was not a separate imitative, but rather an initiative that ran in parallel to other projects within the department. Each resource was expected to implement the tool accurately while still working on projects without limitations.

Quotes from interviewees:

⇨ "Upfront resources are needed."
⇨ "Not able to audit work because of the lack of resources."
⇨ "More resources."
⇨ "QACOE needs more resources to answer questions fast."
⇨ "Knowledgeable full-time resource needed until the software quality assurance tool is implemented."
⇨ "Resources are needed to develop automation."
⇨ "A lot of change and no resources."
⇨ "Because no full-time person was hired, only 10% of the time was spent on the implementation of the software quality assurance tool."
⇨ "Resources are needed to help write automation scripts to for return of investment (ROI)."

Initial Implementation and Communication
Initial Implementation and Communication was the capacity to give out information on the implementation to the individuals within the four departments. The logistics of how accurately information on the implementation itself was given out, managed and communicated.

Quotes from interviewees:
⇨ "The implementation needs to be independent rather then within other projects, it was a nightmare."
⇨ "Should not have implemented using four departments."
⇨ "The implementation was treated like a project rather then as a implementation."
⇨ "The task list for the implementations goals had no dates and expected delivery."
⇨ "Communication within the departments for the implementation was a problem."

Tool Layout
Tool Layout is a major component of the software quality assurance tool system's design. The functional flow in how the system works or of how the system was customized within the company. Therefore, a new application cannot be used properly

unless the user has a working knowledge of its design and how to use it within their department. Managers should be in communication with their employees to ensure that the application is being handled properly, and that all expectations are met.

Quotes from interviewees:

⇨ "Reusing data within the software quality assurance tool was an issue that needs fixed"
⇨ "Cumbersome in reusing automation scripts"
⇨ "Delete functionality within the tool is needed."

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 automatically run functional tests as if they were done manually. 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. Automation issues include infrequent use and complexity of creating automation scripts. 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.

Quotes from interviewees:
⇨ "Unable to use or re-run automation scripts from older contractor scripts."
⇨ "We were not able to automate anything."
⇨ "Did not use automation."

Tool Analysis and Verification
In order for users to accept the system, it must be analyzed and then verified to meet ABC's standards within each of the implemented four departments. If the user is unaware of important elements of SQA tool, or if the users need certain functions added to the tool's existing structure, the software quality assurance tool loses its ability to be effective and used correctly to meet expectations.

Governance & Management
The software quality process and how the software quality assurance testing tool is managed ensures that project testing and the process through which a tool is used meets the targeted timeline and risk assessment. When a user states that Governance and Management is needed, they indicate that minimal control, metrics and audits were utilized during the implementation.

Lack of Upper Management Knowledge of Tools and its Benefits

The departments that implemented the tool believe that upper management does not have adequate knowledge of how the software quality assurance tool is used, nor do they realize the tool's benefits.

Resistance

Areas in which users refused to accept the change of having a new SQA tool caused difficulty throughout the tool's 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.

Company Needs to Listen to Internal In-house Experts Recommendations

The departments in which the tool was implemented believe that management is not using individuals who have expertise and understanding of the tool and how to implement the process to the fullest.

Organizational Layout Issue

The structure of how employees and other resources are organized within ABC's department and corporation structure.

POST IMPLEMENTATION OUTCOMES

As a result of this study, the following issues were observed by the researchers:

- The implementation needed to be a separate activity and not one in which it ran parallel to existing projects.

- The implementation cannot be considered as a project where there is a "drop dead" end-date. Upper management needs to understand that the implementation process is never-ending and always changing, thus the need for constant improvement, awareness and changes with results.

- The tool itself will not create a magical reduction of employee and staff with an increase return on investment, efficiency and productivity. Upper management cannot take the word "automation" to mean that the system will completely manage itself without human intervention. With Software Test Automation every time a change is made to the end product the automation

scripts change with it. The maintenance and training of the "automation" scripts run in-tandem to the upkeep and change of the functional code within the application.

- Reliable, effective, permanent and knowledgeable Software Quality Assurance (SQA) staff is necessary. Business Analysts, Developers and sometimes Project Managers act as Quality Analysts. Very few departments had dedicated QA staff. It was a requirement within each department to have full-time software quality analysts versus a hybrid position or individuals. Dedication is needed along with consistence and knowledge in the ability to be a direct software quality analyst and software tester within each department. Many times individuals within departments would ask "What is QA?", "How do I create a Test Plan document?", and "What are use cases?" Knowledge in both software quality assurance and software testing is fundamental within each department to increase awareness and the company's ability to comprehend and produce quality initiatives.

- The Lines of Business outside of the IT organization needed to support this implementation because they supported the requirements and information that the IT Business Analysts collected.

- Top Management understanding and support. Too much flexibility was given during the implementation. No direct feedback on results was given in a way to show productivity. Very little quantitative productivity metrics were collected.

After the implementation, the manager of the QACOE was transferred to a different department. A new manger was hired from outside the company. No further progress on the software quality assurance tool implementation was made based on this formalized implementation and its outcome. At the time of this study, another new initiative was being recommended by the new manager.

CONCLUSION

The top three issues recorded from the 2008 interview process were Time, Process and Assistance and Training. Comparing these top three current

issues to the issues from 2007 (Process, Assistance and Training, and Tool Implementation Improvements), suggests that Assistance and Training is a major factor in the success for the implementation of software quality assurance tools. What is even more astonishing is that within the 2008 results the number one issue was Time and the fourth issue was Resources. Both of these were key factors for the success of Assistance and Training towards the implementation of software quality assurance tools.

Process was the number-one issue in 2007 and the number-two issue factor in 2008. Process must be aligned if the project is to be successful. As previously indicated, a suitable software implementation process requires a fluid, corporate-wide structure and positive employee communication.

Another finding was the issue of Initial Implementation and Communication. In both 2007 and 2008, it was the fourth most frequently mentioned issue. Employee communication is crucial to having a successful implementation process. If Implementation and Communication is still a problem then the implementation of the software quality assurance tool will not be a success.

When all of the issues are quantitatively reviewed, 2008 displays 145 total issues and 2007 displays 83 total issues. This is an interesting finding, since the formalized implementation was meant to integrate the software quality assurance tool correctly. Also notable are the four new issue categories: Resources, Lack of Upper Management's Knowledge of Tools and their Benefits, Company Needs to Listen to Internal Experts Recommendations, and Organizational Layout Issues. It is possible that more issue topics can arise during a new implementation but the initial thought and goal would be that the quantitative number of total issues would decrease.

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 without providing a basic objective. 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 manger of the QACOE (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 believe the time needed and risks associated with conducting manual testing outweigh the benefits. But Michael Donat [2] indicates that "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 an 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 managed groups. The reason for such a separation is to ensure that no bias exists among different software life cycle entities. In organizations such as ABC that primarily operate under the Mainframe platform, many of the systems are asynchronous when testing. A Business Analyst can conduct testing and documentation – but, as Web-based and Graphical User Interface (GUI) systems become more prevalent, a door is opened to the possibility of punching more holes in the functionality of the system. This created the need for 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 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 many years of being a developer or Business Analyst, but this field should develop employees from inside the classroom rather then 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 and Training is fundamental to any product execution.

The implementation of a formalized software quality assurance tool is a constantly changing process by its very nature. But if top management does not have clear objectives in regard to outcomes of the software quality assurance tool implementation, the process will most certainly never end. Having the correct number of resources, dedicated time, process and initial planning with the appropriate staff is the key to success. Constantly having new initiatives and new individuals managing the implementation does not create the ability to progress but the inability of a company's culture to accept, understand and grow within a software testing and software quality mind-set. The lessons learned and continual change must be understood from mistakes and successes of prior implementations. For the best results, follow this rule-of-priority: 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=281

3. Feldman, S. (2006). Quality Assurance: Much More than Testing. Retrieved February 10, 2008, from Association from Association for Computing Machinery: http://www.acmqueue.org/modules.php?pa=showpage&pid=276&name=Content

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

9. Scarpino, J., & Kovacs, P. (2008). An Analysis of a Software Quality Assurance Tool's Implementation: A Case Study. Journal of the International Association for Computer Information Systems, VOL IX, No. 2, 2008[9].