

**AN ANALYSIS OF AN ENTERPRISE MOBILITY SOFTWARE COMPANY –
MANAGING SOFTWARE QUALITY AND MAINTAINING A COMPETITIVE EDGE
IN FLUCTUATING PERIODS OF CORPORATE GROWTH: A CASE STUDY**

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

*Rhonda G. Chicone, Notify Technology Corporation, rchicone@notifycorp.com & Minot State University,
rhonda.chicone@minotstateu.edu*

ABSTRACT

The Software Quality Assurance (SQA) industry has grown within the last 10 to 15 years, but as the U.S. continues to fight to stay competitive in an increasingly difficult economy, companies must ensure that their processes and products remain top-notch, productive and effective. A persistent problem is that many Information Technology organizations have a tendency to rush into implementing a software quality assurance process without first establishing a viable quality assurance process within each department. This paper reports on data collected in January of 2011 concerning a two-day software quality analysis of Notify Technology Corporation, a Silicon Valley leader in mobile data synchronization and management software systems. The data analysis reveals problems with quality that can arise while growing a mobile software company and explores the changes needed to resolve major issues in the quality process to ensure corporate growth and competitiveness.

Keywords: Software Quality, Software Quality Process, Software Quality Assurance (SQA), Software Quality Process Implementation, Software Quality Management, Mobile Software, Software Development Lifecycle (SDLC), Software Company Growth.

INTRODUCTION

Notify Technology Corporation is an independent software vendor (ISV) specializing in wireless solutions and services that provide secure synchronized e-mail, calendar, contacts and tasks (PIM) access and management to any size organization or business on a variety of mobile device platforms and networks. These include Apple iOS, Research in Motion BlackBerry, Google Android, HP webOS, Microsoft Windows Phone 7 and Nokia Symbian. Presently, Notify Technology Corporation has created three enterprise based software products. They are NotifyLink, NotifySync and NotifyMDM. Notify Technology Corporation sells their products in both domestic and international markets.¹

It is crucial that the development of all mobile and software systems adhere to high quality standards in order to guarantee that bugs and defects are acknowledged and eliminated before the application is implemented. Van Genuchten & Hatton (2010) state that software production is booming (especially with Internet applications like social media networks and online video) even though the quality processes needed to create reliable software are notoriously time-consuming and increasingly expensive. In 2010 alone, the number of lines of source code for an average mobile device was estimated at 10 million.² This statistic demonstrates the exponential growth of the mobile / smartphone market in response to the consuming public's increasing desire for personal devices that can download and share information in the blink of an eye. In fact, smartphone technology has developed at a faster rate than any other computer technology.³

With such fast-paced online and mobile media growth, many of today's software developers and organizations feel they can't keep up with Quality Assurance (QA) related processes – yet, it is at times like these that quality management is imperative for success.

According to Feldman (2006), Software Quality Assurance (SQA) is not just a nicety – it's a necessity that requires patience, dedication and guidance throughout its layers of testing and procedures.⁴ But organizations that do the right thing by infiltrating the Software Development Life Cycle (SDLC) with SQA will reap the most benefits.⁵ 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. In fact, the Handbook of SQA indicates that SQA is “the functional entity performing software quality assessment and measurement.”⁶ Similarly, the IEEE Standard 12207 defines Quality Assurance 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.”⁴ Hence, quality is essential and companies must be sure to invest in both process and testing – not just one or the other.

For Notify Technology Corporation, it was very apparent that the company had a problem; the company was growing but its QA department and overall mindset regarding “quality” was stagnate. The QA department was focused solely on software mobile *testing* rather than the *assurance* of Notify’s processes and products – what was supposed to be a well-rounded QA department was, in essence, only a software testing department.

A Researcher and QA expert conducted a two-day, comprehensive analysis of Notify Technology’s software quality assurance processes and procedures. Knowing that “quality” is only as good as how its processes are implemented across the company, the Researcher met with Notify Technology Corporation’s QA department and every department that the QA personnel worked with. The results of the analysis suggest that the problem was not rooted in the QA department (i.e., software testing department) alone, and that the culture had settled over the entire enterprise.

Shoenberger (1982) outlines five rules that Japanese total quality control emphasized in the 1980’s, which are still very true today:

1. A goal of continual quality improvement, project after project.
2. Worker (not Quality Control department) responsibility.
3. Quality control of every process, not reliance upon inspection of lots for only selected processes (defect prevention, not random detection).
4. Measure of quality that are visible, visual, simple and understandable, even to the casual observer.
5. Automatic quality measurement devices (self-developed).

All five of these points are important take-away lessons for companies struggling with quality. In greater detail, a company must first set goals for quality improvement and it must ensure that quality is maintained after a product and/or project is completed. A single worker or department is not responsible for quality – in Notify Technology Corporation’s case, quality rested on the software testing department’s shoulders, but every department must do its part to ensure quality. Quality must be easily recognizable by the end user; Notify Technology Corporation’s mobile software based systems should be easy to use and contain good quality attributes. And lastly, quality needs to be self-engineered with a system of checks and balances to maintain quality during use. That is, the employees and respective departments of the organization which creates the product need to have an understanding of the correct processes and procedures, working together as a team to ensure statistical measurements and changes.

This paper reports on data collected in January of 2011 during a two-day software quality analysis. The information was gathered through open-ended interviews with employees at Notify Technology Corporation, a Silicon Valley leader in mobile data synchronization and management software systems. The data analysis reveals problems with quality that can arise while growing a mobile software company and explores the changes needed to resolve major issues in the quality process to ensure corporate growth and competitiveness.

DATA COLLECTION

The data collection and research process performed by the Researcher was similar to the method outlined in the journal article, *An Analysis Of A Software Quality Assurance Tool’s Implementation: A Case Study* (Scarpino & Kovacs, 2008).⁵ The research performed in 2008 “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.”

Data for this research paper was gathered during 15 interviews with four departments over the course of two days in January 2011. The four departments at Notify Technology Corporation were 1) QA Department; 2) Software Development Department, 3) Technical Support Department, and 4) Technical Writing Department. The organizational structure of these departments was as follows: the QA department had one manager and three supervisors; the Software Development department had three managers and three project leads; the Technical Support department had three managers; and the Technical Writing department had one technical writer. The researcher presented the completed report to the Chief Technology Officer of Notify Technology Corporation at the end of the second day.

This study’s data was collected through six departmental interviews. The method included open observational analysis and documentation – each time an interviewee mentioned an issue it was documented and assessed by the researcher.

Interviews were conducted as follows:

- Interview 1 – One software test “QA” manager
- Interview 2 – Three software test “QA” supervisors
- Interview 3 – Three software development managers
- Interview 4 – Three software development project leads
- Interview 5 – Three technical support managers
- Interview 6 – One technical writer

The topics covered during these six interviews included Business Process Optimization, Defect Management & Control, Governance, Knowledge Management, Metrics & Dashboards, Post Installation, Product Installation, QA Environment, Release Schedules, Review and Inspection, Usage of SDLC, Test Audit, Test Automation Techniques, Test Kickoff, Test Planning & 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.

FINDINGS

Software Testing “QA” Department

Issue 1 – Test Case Steps

The test manager and the team should break up the test cases and provide more detail within each. The company currently has test cases with more than 1,600 steps combined in small groups, but it would be more efficient if there were an increased number of test cases with fewer steps. The reason why this occurred in the testing group was due to the lack of requirement structure coming from the functional specifications. For example, the testing group was trying to test as much as they could in a short amount of time and with very little information. A good rule to operate under is that test cases should have no more than 25 steps, maximum. This will assist with any relation to the requirement / functional specs that could be changed for new functionality. So when the test team sets up the test cases, they need to be related back to the functional spec layout (i.e., requirement document).

Issue 2 – Test Case Layout

The layout of the QA department’s test cases need to be related to the functional specifications (i.e., requirements) versus a list of regression scripts. So, as indicated in Issue 1, the resolution is to break the test cases into functional areas of the application so that Notify Technology Corporation can tie in those areas to any functional change or test case maintenance in the future.

Issue 3 – Analysis of Test Activity

Regardless if the “QA” department does or does not have requirements / functional specs adequate enough to test against, the software testing department needs to take the list of functional changes or additions and analyze it with the test group. The software testing department needs to look at areas where the new functionality has a possibility of being impacted. The software testing group should come up with a list of “what if’s” or possible situations that could occur with the initiation of a new piece of functionality or change. It is best to outline everything and bring it to the attention of the software development team in a physical meeting. The QA department was using a source document control system and issue documents sending e-mails to communicate instead of using face-to-face meetings. The QA department needs to have an actual in-person conversation with the development managers and/or developer.

Issue 4 – Functional Specifications / Requirement Document Analysis

The testing group, or at the very least a testing supervisor, needs to be involved from the beginning when a new change or functionality has been initiated from management to software development. The QA group needs to be aware of the details of what is to come – not just a list of functionalities.

Issue 5 – Efficiency of Testing

Each time the QA group conducted a test they used a full regression method from “A to Z.” Testing all of the functionality over-and-over to ensure that everything is risk-free and changes were not made inadvertently is understandable but it takes a lot of time and resources. The software testing department needs to cut down and analyze areas of risk – and from these areas of risk, test the systems or mobile application functionality. Areas of constant repetition need to be automated – these are functional areas that have little change in development over time but still need to be quality-assured on a regular basis. The software testing department needs to also keep in mind that they should not only conduct “positive testing” (the likelihood of what should happen if the tool is used properly) but also “negative testing” (the likelihood of what may happen if the tool is used improperly). Negative tests are typically issues that are hard to replicate, thus the reason why Technical Support Department may receive calls from the customer. The solution is to test using more mobile software device simulations, in combination with software test automation, rather than the actual physical devices.

Issue 6 – Test Plan and Defects are Knowledge Bases

The QA department needs to have the test cases and defect repository used in a way to act as if they are knowledge databases to other departments. Notify Technology Corporation’s Technical Support indicated that they typically do not use Bugzilla (the bug tracking tool), nor the test case repository, which are both managed by the software testing department .. It is of utmost importance that the QA department enters into communication with Technical Support Department to note the risk of not using these tools, and give the Technical Support Department access to each in case they need them during a current or future project.

Issue 7 – Test Case and Defect Traceability

The Test Case Repository that is being used by the software testing department was created in house and its sole purpose was to document all test cases. They need to keep using the Test Case Repository system but to make sure that they can architect each of the test cases in functional areas as indicated above – no more than 25 steps. The test case needs to be traced back to the requirements (and vice-versa) and obtain either a “pass or fail” status relating the defect. It seems that the QA department is doing this to some degree, but not in much detail. They need to make sure that all requirements / functional specs are documented in detail, relating their respective test cases (both positive and negative) to a list of executions and to the defect itself. The QA department needs to be able to clearly trace the list of defects to the test cases and the requirements / functional specs themselves. Both Bugzilla (for defect tracking) and the Test Case Repository are a good start, but they are being used too informally by not fully “talking” to each other and the functional specs. Other software systems exist to help create test case-to-requirement /

functional specs-traceability, which can tie-in the requirement to the test case, execute both manual and automated tests and tie it to a defect.

Issue 8 – Status “Closure”

The team that assures the defect repository, Bugzilla, is the QA team. Thus, the development team should not close any defect after it is released in production. It is the responsibility of the QA team to make sure it is working after it is released in production, and then to close it. A software development manager is currently closing the status in the Bugzilla defect-tracking software. The QA manager needs to be doing this.

Issue 9 – Test Training

The QA department needs to be trained on how to analyze and conduct various testing techniques. It is recommended by the researcher that Notify Technology Corporation sends a few of its software testing members to attend some non-vendor-specific conferences in order to gain a better understanding how to conduct testing. The members attending these conferences and training sessions need to make sure they reach out to peers at the conference to delve deeper into their experiences. The current software test manager is doing the best he can with the knowledge and experience that he has obtained to this point. He just needs a bit of guidance from industry experts. The software testing members within the QA department need to learn how to work as a team, rather than independently, to analyze the impact of the functionality and create pieces (rather than large chunks) of it for the test cases. The smaller pieces are easier to analyze, maintain and conform to the requirements / functional specs. Once members of the department take a few software testing training courses, they should have a much better understanding of how to format each test case and its testing approaches to mitigate software risks.

Software Development Department

Issue 10 – Functional Specifications / Requirements Document

Functional specifications / requirements documents should never be created by the person who develops the product; doing so opens up a door for the developer to also manage the direction of risk and change. This is precisely what Notify Technology Corporation is allowing to happen. The Researcher deduced from the development and QA teams that changes tend to occur at the last minute and it is often difficult for the developer to communicate with the tester about the changes. This is one reason why Notify Technology Corporation employs many testers to protect against the risk of the “unknown,” which is the functional specifications / requirement document. The software testers often are unaware of changes coming down the pipeline and they feel their project time is cut when they are given a list of functionalities at the last minute. They need enough time to analyze the list, make test requirements from it, make test plans from the test requirements and test the one piece of functionality from the list. If Notify Technology Corporation had a Requirements group (which did not exist at the time of this study and was recommended by the Researcher) that “followed” the product from management to development with the software testing group by its side, development would be relieved from any biased control. It would also decrease the risk and give more time for both the development and the QA teams to do their jobs.

The Researcher had previously observed environments similar to Notify Technology Corporation’s, where development is one of the smallest groups but technical support and testing are some of the largest. This is usually because the teams are trying to fix defects well after the product goes live – defects that, with the proper oversight, could have been caught much earlier. Notify Technology Corporation needs at least three people managing current and future mobile software requirements: one as the lead, another for server management software functionality, and a third for devices. In Notify Technology Corporation’s case, these three individuals should come from Development since this department is conducting these activities anyway.

Technical Support

Issue 11 – Technical Support

Technical Support uses an employee from software development as the “go-to” person for resolving customer issues. It would be in Notify Technology Corporation’s best interest to develop an additional “Software Tech Support” group. This is yet another example which illustrates that development is taking on too much – they are supporting both the software specification writing and the technical issues coming from technical support. If another Technical Support group is formed, the point-person from development who currently assists software support could serve as an example; he is a natural post-production individual who can create timely resolutions for customers.

Issue 12 – Communication

As with everything else, communication is key. Technical support communicates with development but does not communicate with the software testing department. It is imperative that the QA team be a part of the analysis and troubleshooting process when an issue comes up in production. As observed by the Researcher, the QA group does not count analysis of defects (or trying to figure out ways to “break” the tool) as a part of their responsibility; they test by regression and only positive testing, not negative. Also, development closes all defects in production post-release, which should be done by the software testing department to ensure that all open defects are verified and closed in production. More effort should be made to keep the software testing department informed of the full functionality in the production environment.

Technical Writing

Issue 13 – Working with a Requirements Team

The Researcher observed that the technical writer is doing a great job. Her forte is composing documents into guides and help-files for the clients. However, she does not have a background in Software Requirements engineering. It is suggested that one of the development managers become a business requirements lead who will interface with development and software testing, with the technical writer and technical support team in supporting roles. The technical writer should therefore keep her role as-is for the most part, but integrate within a software requirements department since the technical writing is part of requirements development. The technical writer can then start growing their career and expand as a requirements engineer while maintaining technical writing roles.

Additional Analysis

Issue 14 – Time

Every group that was interviewed over the two days indicated that “time” was an issue, except for Technical Support and the technical writer. The reason for this may be because each group’s work is contingent upon factors largely out of their control, and therefore they find it hard to prepare themselves before the release occurs.

Issue 15 – Disconnect

A disconnect exists between the QA team and development group. Based on the Researcher’s observation, it seems that there is “task fence,” or “political totem pole,” within Notify Technology Corporation that prevents these two teams from interfacing regularly. Removing development’s responsibility for functional specifications may help dissolve this barrier, but both sides need to make more of an effort to build a bridge and meet in the middle. Enjoyable teambuilding exercises and other informal activities may be a good way to initiate contact and get everyone interacting on a personal level, as human beings – not just as co-workers. For example, the development group could plan a weekend bike ride for both groups. In return, the QA group could plan a golf or movie outing for both groups. A better connection between these two departments starts with treating each other as equals, with less ego about who does more work, who is more reliable, who is more technically savvy, etc. Breaking the disconnect

before it grows larger will improve the culture and minimize tension. The Researcher's previous analysis at other organizations has demonstrated prohibited growth when internal politics gets to be too much for employees to bear.

Issue 16 – Product Manager / Delivery Manager

A Product Manager or Delivery Manager (a non process-oriented Project Manager type) should be integrated within the organization. The atmosphere at Notify Technology Corporation is very customer driven and deadline-driven and updates on progress are made, but the details of what happens within Development, Test, Technical Support all passes through without a formal understanding of why things are being done the way they are. Having a Product Manager take care of this will also allow more time for the Chief Technology Officer to align her research with the growth of the company. The Researcher recommends reviewing some GANTT Chart software to help describe and communicate these deadlines.

Agile vs. Agility

Agile is working in small groups to develop the task at-hand in sprints with minimal documentation. Agile software development employees work in small groups. They physically work in the same area, basically eye-to-eye to one another, releasing their deliverables in small spurts. The Researcher believes that Notify Technology Corporation may need a little more formal project direction. This should not come from a Project Manager who tends to be heavy on process, but rather from a Delivery Manager who can make quick resolutions and act as the communicator between groups instead of just giving deadlines and status updates. While Notify Technology Corporation indicates it is following "Agile" development, it is actually misusing this software development process term to mean it "we are following our own process with company creativity and *agility*," versus an *agile* process.

Issue 16 – Metrics and Governance

There are currently no benchmarks or metrics against which to measure how well software testing, quality assurance and software development were performed for current projects over previous ones. The Researcher witnessed a "test and test again until it looks right" mentality, which included a "keep developing without resolving many of the issues" and "add more bodies as the number of problems increases" performance structure. In order for Notify Technology Corporation to grow, proper quality assurance and measurement is absolutely key. The company can still be quick in delivering the product on time, but more effective and efficient. The Researcher suggests that Notify Technology Corporation conducts "lessons learned" meetings with management and at least one or two individuals each in the QA, Development and Technical Support groups, including the product hosting environment manager and tech writing. A note taker in these meetings should write down all of the items discussed in detail, such as: "What was the largest challenge the project faced? How did your departments come together to solve it? How could they work together better in the future? What will you do if the same issue happens again?" The more Notify Technology Corporation understands what went wrong in older projects and how its employees worked through it to solve the issue, the more the organization will grow.

Issue 17 – Research is an Important Contribution to Company Growth

The Researcher's findings indicate that the Chief Technology Officer should research new tools and/or methods to implement across the organization and that she review them on her own, at least at first. The Researcher was concerned that if the CTO were to ask Development to do this kind of research, they would be resistant to change and only do the bare minimum, purposely putting up roadblocks that could hinder growth. The Researcher has faced this at other organizations, and is confident that the CTO knows the products better than anyone since she is responsible for all product aspects including definition, architecture and driving the product schedules. She also knows what she intends to make of Notify Technology Corporation's future.

CONCLUSION

While assessing the company's software quality during his two-day on site visit to Notify Technology Corporation, the Researcher's findings point to the need for many changes in the short-term, and more changes to follow later on.

Change 1 - QA Training

The QA Manager will attend a Software Quality Assurance conference at the end of March. In addition, he will make of an effort to connect with peers from like companies to share ideas and solutions to common QA problems. This knowledge will be shared with the software testers at Notify Technology Corporation.

Change 2 - Customer Problem Escalation Process

The escalation process has been revamped. The old process was to escalate a customer reported software defect to the Software Development Department. The new process is to escalate it to the QA Department. The QA Department will try to duplicate the problem and document all conditions. This off loads the software development staff and improves the QA Department's real-world test cases. After a defect has been escalated the QA Department will then determine if the defect is a setup or configuration problem or a bug. If it is a bug then the bug will be logged in Bugzilla for further analysis by the Software Development Department.

Change 3 - Requirements Engineer

A Requirements Engineer position was created with the sole responsibility of working with management and creating the Requirements / Functional Specifications for the software products. This responsibility will no longer be a Software Development function.

Change 4 – Communication Improvements

Communications will be improved in several ways. First, with the advent of the Requirements Engineer, in time, the barriers inhibiting the communications between the software developers and testers will improve. Second, even though Notify Technology Corporation has team building exercises within its respective departments, team building exercises will occur between departments. Third, software developers and testers are having more face-to-face meetings analyzing functionality and measuring risks.

Change 5 – Bug Closing

Closing bugs found in Notify Technology Corporation's software products will no longer be done by the software development manager but rather the QA manager.

Change 6 – Agile Software Development Process

The current software development process at Notify Technology will slowly change to an agile process. The software development managers will receive training in this area which will include an off-site conference.

Change 7 - The Creation of Publications Department

A publication department was formed to include the Requirements Engineer and Technical Writer. In addition, another Technical Wirier was hired.

Change 8 – Director of Product Development

In time, the CTO will hire a Director of Product Development. This will off-load the CTO so she can do research and spend more time working on future product plans for the company.

Notify Technology Corporation wants to continue to be a market leader in creating mobility based enterprise software systems. To do so the CTO realizes changes must occur to reach the company's goals. To remain successful and to grow, Notify Technology Corporation's software products must be effective, efficient and reliable to its enterprise customers. Continued success cannot occur unless software quality is assured.

REFERENCES

1. *Notify Technology Corporation*. (2011). Retrieved from <http://www.notifycorp.com/about/index.htm>
2. Van Genuchten, M., & Hatton, L. (2010). *Software: What's In It and What's It In?* (pp. 14). IEEE January/February 2010.
3. Papows, J. (2011). *Glitch The Hidden Impact of Faulty Software*. (pp.101) New Jersey: Prentice Hall PTR.
4. 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>
5. Scarpino, J.J. & Kovacs, P.J. (2008). *An Analysis Of A Software Quality Assurance Tool's Implimentation: A Case Study*' *Issues in Information Systems VOL IX, No. 2, 2008*
6. Schulmeyer, G., & McManus, J.I. (1999). *Guide to Software Quality Assurance*. (pp.7) New Jersey: Prentice Hall PTR.
7. Schonberger, R.J. (1982). *Japanese Manufacturing Techniques: Nine Hidden Lessons in Simplicity*. (pp. 7) New York: The Free Press.