

THE QUALITY OF AGILE – TRANSFORMING A SOFTWARE DEVELOPMENT COMPANY’S PROCESS: A FOLLOW-UP CASE STUDY

*John J. Scarpino, Robert Morris University, scarpino@rmu.edu
Rhonda G. Chicone, Kaplan University, rchicone@kaplan.edu*

ABSTRACT

The Software Quality Assurance (SQA) and Software Testing (ST) in industries have grown within the last 15 years, but as the U.S. continues to grow in an increasingly tough and competitive economy, companies must guarantee that their processes and products remain top-notch, productive, and effective. As we recently saw with the Health Insurance Marketplace website, software testing and quality assurance is a large part of planning and engineering within software development to ensure functionality, scalability, performance, and security.²

*A stubborn problem is that many Information Technology establishments 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 on August 23, 2013 from a single day follow-up of a software quality assurance analysis (originally conducted in January 2011) at Notify Technology Corporation, a Silicon Valley leader in mobile data synchronization and management software systems. The results from the 2011 case study has already been published: *An Analysis of an Enterprise Mobility Software company-Managing Software Quality and Maintaining a Competitive Edge In fluctuating Periods of Corporate Growth: Case study.*⁸*

This follow-up study and data analysis reveals changes and emerging trends that arose as a result of the first case study. Agile development practices were a major piece of implementation and the focus of this follow-up study, which has led to many updates and thus the need to report and analyze again. While quality assurance issues can arise while growing any software development company, this paper explores the positive and negative changes that have occurred since the previous analysis in 2011.

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

INTRODUCTION

Notify Technology Corporation is an Independent Software Vendor (ISV) focused on wireless solutions and services that deliver secure synchronized e-mail, calendar, contacts, and tasks (PIM) access and management to any size institute or business on a diversity of mobile device platforms and networks. These comprise Apple iOS, Research in Motion (RIM) BlackBerry, Google Android, Microsoft Windows Phone 7/8, and Nokia Symbian. Notify Technology Corporation has created three enterprise based software solutions. They are: NotifyLink, NotifySync, and NotifyMDM. Notify Technology Corporation sells their products in both domestic and international markets.⁶ As of late 2013, right after this study was conducted, the company’s business and assets were acquired by Globo Plc. Globo Plc. was created in 1997 and is managed through head offices in New York, London, and Athens.

It is vital that the development of all smartphone, mobile, and software systems follow to high quality principles in order to guarantee that issues and defects are recognized and eliminated before the application is implemented. Van Genuchten & Hatton 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.¹³ This indicator demonstrates the exponential progression with the use of mobile and smartphone market in response to the intense public craving 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.⁷

With such change in the online and smartphone progress, many of today's software developers and establishments feel they cannot keep up with Quality Assurance (QA) related methods – however, it is at times like these that quality management process and proper software testing are imperative to success.

According to Feldman, Software Quality Assurance (SQA) is a necessity that involves patience, dedication and direction during 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 see the most payback.⁹ SQA includes the complete SDLC, which comprises processes such as software design, coding/implementation, source code control, code reviews, configuration, and testing at both the development- and user-end, and change and release management. SQA also includes software testing, which encompasses functionality testing, regression testing, load and performance testing, and security testing. SQA not only makes certain that an application is relieved of faults and defects, but that it is dependable, fully predictable, supportable, and completely functional. In fact, the Handbook of SQA indicates that SQA is “the functional entity performing software quality assessment and measurement.”¹⁰

Change Recommendations from the 2011 Case Study

For Notify Technology Corporation, it was very apparent that the company had problems, as reported within the 2011 case study. The company was growing but its QA department and overall mindset regarding “quality” was stagnant. The QA department was focused solely on software mobile *system 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. Here are eight recommended changes that needed to be made, as concluded by the 2011 case study: 1) QA Training; 2) Customer Problem Escalation Process; 3) Requirements Engineer; 4) Communication Improvements; 5) Bug Closing; 6) Agile Software Development Process; 7) The Creation of Publications Department; and 8) Director of Product Development.

As the 2011 research notes, the Researcher/QA Expert knew that “quality” is only as good as how its processes are implemented across the business. In the follow-up study, the same Researcher/QA Expert met with Notify Technology Corporations' QA department and every department that the QA personnel worked with. The intention of this was to identify the source of the issue, rather than suggesting the QA department was the sole problem. A Researcher/QA Expert conducted the previous case study in two days; this follow-up study was to conduct similar research as open-ended interviews with people who are in the same positions that were interviewed in the past. The purpose was to see if changes had occurred based on the recommended eight changes. One of the biggest areas that the case study in 2011 suggested was Change 6 – Agile Software Development Process.

Agile Software Development

Agile software development was not implemented within Notify Technology until after the 2011 study. This paper will describe how agile impacted the company. Based on the research, the software process model being used by Notify was similar to a waterfall process, which required each step be completed before beginning the next. This is much different than agile. According to an IEEE article titled *What Do We Know about Agile Software Development?*, the agile methods remedy the issues of an unpredictable world, focusing on the value of competent people and what their relationships bring to software development.³ Table 1 ‘Traditional and Agile Perspectives on Software Development’ highlights the differences outlined in the article.

Table 1. *'Traditional and Agile Perspectives on Software Development'*

	Traditional View	Agile Perspective
<i>Design process</i>	Deliberate and formal, linear sequence of steps, separate formulation and implementation, rule-driven	Emergent, iterative and exploratory, knowing and action inseparable, beyond formal rules
<i>Goal</i>	Optimization	Adaptation, flexibility, responsiveness
<i>Problem-solving process</i>	Selection of the best means to accomplish a given end through well-planned, formalized activities	Learning through experimentation and introspection, constantly reframing the problem and its solution
<i>View of the environment</i>	Stable, predictable	Turbulent, difficult to predict
<i>Type of learning</i>	Single-loop/adaptive	Double-loop/generative
<i>Key characteristics</i>	Control and direction Avoids conflict Formalizes innovation Manager is controller Design precedes implementation	Collaboration and communication; integrates different worldviews. Embraces conflict and dialectics. Encourages exploration and creativity; opportunistic. Manager is facilitator. Design and implementation are inseparable and evolve iteratively
<i>Rationality</i>	Technical/functional	Substantial
<i>Theoretical and/or philosophical roots</i>	Logical positivism, scientific method	Action learning, John Dewey's pragmatism, phenomenology

Lan Cao and Balasubramaniam Ramesh describe four areas that are impacted by agile methods: Environment, Values, Beliefs, and Implementation of practices.¹ This information is outlined below:

Environment – Agile methods focus on system development in a dynamic environment with volatile requirements, changing technology, and critical time to market. Traditional methods are more appropriate in a relatively stable environment in which quality is the major concern.

Values – As specified in the “Manifesto for Agile Software Development” (<http://www.agilealliance.com>), agile methods value interaction, collaboration, and adaptability; while traditional methods value planning, predictability, high assurance and control.

Beliefs – Agile methods stress that requirements emerge throughout the product development cycle. Traditional methods demand complete and accurate requirement specification, preferably before development. Also agile methods presume that change is unavoidable and should be embraced, while traditional methods strive to control change.

Implementations of Practices – As a result of the previous differences, the implementation of software development practices differs for agile and traditional software development, even though each practice has a long history in software engineering. Agile development uses some practices to the extreme compared to traditional software development. For example, although most packaged software is delivered in releases, agile software development tends to have several and more frequent releases.

The agile method is much more “open” to frequent changes and discussions during scrum meetings, and not required to have stringent documentation. As an article by Julia Sennikovski stated, when using agile as compared to the waterfall SDLC, it helps “to shorten the duration of the test project, and ultimately achieve an earlier product release date ... [Julia] introduced the practice of dividing large features into smaller standalone subsets and negotiating early deliverables to test for these subsets. As a result, the test team was able to begin testing earlier and shift focus to the newly completed features right away.”¹¹

According to an article titled *Assumptions Underlying Agile Software-Development Process from the Journal of Database Management*:¹²

It is important to be aware that agile development approaches are built on many, possibly [unspoken], assumptions, and that these assumptions are probably not appropriate for all organizations or development projects. When the assumptions made by agile development methods are not in alignment, or even direct

conflict, with those of the organization, managers in charge of development need to take steps to adapt the agile development process if such an approach is adopted, or be confident in choosing a traditional approach, knowing that it will better fit their environment. If this is not done, an agile development approach may very likely provide less-than-desirable results because of the limitations that result from these assumptions.

This paper reports on data collected on August 23, 2013 during a follow-up study which was based on previously published research conducted in January 2011 during a two-day software quality analysis. The information within this second case study was gathered in a similar fashion to the previous research. Open-end interviews were conducted with employees at Notify Technology Corporation, a Silicon Valley leader, about smartphone use, mobile data synchronization and management software systems. The purpose of this data analysis is to see if the eight changes from the previous study were put into effect, and to identify any additional problems with software quality assurance within this growing mobile software company.

DATA COLLECTION

The data collection and research process performed by the Researcher was similar to the method outlined in the prior study, as this paper acts as a follow-up to *An Analysis of an Enterprise Mobility Software company-Managing Software Quality and Maintaining a Competitive Edge In fluctuating Periods of Corporate Growth: Case study (Scarpino & Chicone, 2011)*.⁸

The original research performed in 2011 was gathered during 14 interviews within 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.

First Analysis

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; and Interview 6 – One technical writer.

The research performed in this follow-up study was conducted throughout a single day on August 23, 2013, with seven departments with 10 interviewees. The seven departments at Notify Technology Corporation were 1) Scrum Manager (new role since last analysis); 2) Technical Support; 3) VP of Software Development and Quality Assurance (new role since last analysis); 4) QA Department; 5) Product Manager (new role since last analysis); 6) Software Development; and 7) Technical writer. The following positions were newly added as the Chief Technology Officer of Notify stated as these were part of the recommendations from the first analysis study: Scrum Manager, VP of Software Development and QA, and the Product Manager.

Follow-up Analysis

Interviews were conducted as follows: Interview 1 – One Scrum Manager; Interview 2 – One Support Manager; Interview 3 – One VP of Development and QA; Interview 4 – One Software Test “QA” manager; Interview 5 – Two software “QA” supervisors; Interview 6 – One Product Manager; Interview 7 – Two Software Developers; and Interview 8 – One Technical Writer.

In both studies, the researcher presented the completed report to the Chief Technology Officer of Notify Technology Corporation, who is the second author to this paper.

This follow-up case study, like the previous study, was collected through departmental interviews. The method included open observational analysis and documentation – each time an interviewee mentioned a positive change or issue it was documented and assessed by the researcher.

The topics covered during these six interviews included Business Process Optimization (BPO), 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 mentioned by the interviewees were documented and intermittently interviewees were drawn into additional conversation after they responded, so as to improve documentation.

FINDINGS

Below are the actual findings from each of the interviews.

Scrum Manager

Note: A recommended change from the first study was to add “Agile Software Development Process” within the company’s software development process, which required a Scrum Manager. The Scrum Manager was hired from within the company, and this person was a QA Manager who participated in the previous study.

Positive Change – Scrum Process: There are now fewer gaps between corporate management and ownership due to having a VP of Development and Quality Assurance; this, along with the addition of a scrum process made the process much more effective and efficient. Retrospectives, backlogs, and stand-up techniques used by the scrum methods made the projects more successful. The escalation process is quieter and easier to accomplish. The scrum process improved the SDLC (Software Development Life Cycle) process and it is essentially “self-healing.”

Positive Change – Communication: The new process has helped various departments and individuals work together better and open lines of communication. Friendships are now being developed and a sense of teamwork is being embedded into the Development and QA teams; in the past, these groups acted like enemies. Release schedules are well-communicated and now trying to improve functionality rather than simply meet a deadline.

Positive Change – Quality: The company is now conducting sprint planning for each release. Also, code reviews are occurring, which is resulting in increased quality. Recommendations are also being made without being mandated, which was not the case in the past. Test Automation is being developed and used for nightly builds. Test Planning and Design, and Test Requirement Reviews are also being implemented and naturally occurring within the agile process. Overall, the quality has increased from the time of the previous analysis.

Positive Change – Culture: Culture was positively impacted by the agile process, which has created an environment that fosters teamwork. This process has made the employees happier and the experience of working at the company more enjoyable, which positively impacts the outcome of the projects.

Support Manager

Positive Change – Scrum Process: By having a scrum methodology, the employees were informed of what had been changed for each sprint. They were more involved and knowledgeable about the processes, and this assisted their communication with customers. Defect management and control performed better, because agile was faster, more effective, and more efficient. After installation of the agile process, new product release meetings were held with the team. The speed of the agile process was especially effective for a two-week turn around; defects were fixed and resolved faster.

Positive Change – Communication: Communication has improved with the customer simply by using the scrum process. The QA team is able to explain information to the client and bring up client issues during scrum meetings to make more tactical decisions. From a release-schedule perspective, the Technical Support team now understands what is being released and how it impacts both the systems and the customer.

Positive Change – Quality: The scrums added more information about how the systems could be impacted. Also, it created better quality and faster delivery to the customer. In the past, waterfall process releases to the customer would occur without the QA and Testing departments' involvement; now QA tests everything before it is released.

Issue – Versioning and Quick-Release Cycle: It is very hard for groups to keep up with the current version because of how fast new ones are created. Technical Support is negatively impacted by this and can see how it also negatively impacts QA.

Issue – Requirement Review: Requirement reviews need to occur more often with all of the teams, including the Technical Support team. Often, releases occur and Technical Support is unaware of the release's breadth of functionality.

Issue – No Scrum Involvement: Technical Support needs to be involved with all scrums so that they are aware of what is being worked on for future development. At the time of deployment, Technical Support must be ready to provide customer support. Adding the Technical Support team to the scrum meetings as often as possible would create more effective customer management.

VP of Development and QA

Note: A recommended change from the first study was to add "Director of Product Development" to the company's structure. Thus, the CTO promoted one of the prior study's Development Managers to VP of Software Development and QA, and made him in charge of both software development and quality assurance.

Issue – Bias: In the past, the VP of Development and QA was not in charge of QA, and the QA and Development teams were constantly battling with each other. Now that both teams are under a former development manager, both teams are working together much better. It is very important that no bias exists within the company and everyone's work effort is put toward the company goal rather than individual roles.

Positive Change – Scrum Process: With the scrum process, the formalization of the requirement and traceability document was created to assist the development process, continued process improvement, and proper functional developmental structure. Development functionality is now decided and agreed upon, whereas in the past it was a top-down decision from the CEO. Defect management and control is now driven by development to understand what features and functionalities need to be fixed. Product installation is now more productive since the sales team is also aware of what is being released; no surprises exist. With scrum, development and QA want to work together. Also, with scrum and because of more collaboration, nightly builds became a success whereas in the past it was very slow to approach.

Issue – Requirement Needs Improvements: The documentation process for requirements are not in a state where it makes development and QA successful. User Interface (UI) design wireframes, user stories and use-cases need to be in much more detail.

Issue – Internal Training and References: Internal education needs to be available for resources needing education and reference support. For example, areas for support and reference include: project management, agile development and scrum techniques, requirement document creation, and traceability techniques.

Issue – Metrics and Dashboards: The ability to quantify quality, governance, and maturity is not being done. This ability needs to exist in order to benchmark current standing for any future changes. No one is ensuring overall company success.

Issue – Professional Trainer for Customers: Specified trainers should be appointed for the products in order to improve customer interaction and the customer's knowledge of how to use the products. The technical support department can then utilize the customer's knowledge to build a more successful solution to better meet the needs of the customer.

Positive Change – Communication: Sprints and scrums are now directing communication within the groups. Groups have a better understanding of what is coming down the road rather than simply being brought up at the last minute. Also, in the past sales communication and information was not heard; now within the scrums, information from the sales team is being understood. Development and QA are now finally communicating within scrums.

Positive Change – Quality: Quality exists more now because of the development and QA interaction. The QA and Testing departments now have fewer people, but test automation has been in development and improved to reduce the need for “bodies” and created a better effective and efficient process. These automation testing techniques also assist with the overnight build releases.

Software Test “QA” Manager

Note: This software Test “QA” Manager was hired from within the company, and was a Technical Support Manager in the prior study.

Positive Change – Scrum Process: Agile requirements are now understood from the beginning, which assists in understanding the user stories and functional process of what is being developed. Deadlines are now being made and met up-front because of the scrum meetings. In the past, more people worked in QA and this was especially a need because waterfall was being used; now with agile there is less of a need in the QA group. Agile is working a lot better as SDLC than waterfall.

Positive Change – Communication: The communication of release schedules exist now; in the past it was non-existent. The test workflow is now communicated within the scrums, making everyone part of the QA and testing processes. The user stories make it much easier for other groups to understand testing approaches and techniques.

Positive Change – Quality: Issues and defects are now being caught early since the QA team is aware of the functionality that is being developed and released. Also, the team implemented more test automation. This has resulted in an increased ability to test more with fewer QA resources. Having the proper testing tools and testing technique within agile is key for quality success.

Software “QA” Supervisors

Issue – More Test Automation Needed: Many QA resources were laid off, so this means that more automation is needed to ensure proper functional test coverage.

Positive Change – Scrum Process: Agile assisted with the Business Process Optimization (BPO) and understanding of how functionality and development impacts the system, where internal processes do impact the customer and overall business at Notify Technology Corporation.

Positive Change – Communication: Release schedules assisted with communication. This helped QA and Testing Departments and the other departments stay on the same page. Another part in how communication was resolved was with change management. If needed, anyone can create a build for the functional release.

Positive Change – Quality: Software test automation is now being developed and used. This was not the case in the past analysis. Test automation resulted in having more coverage over all of the functionality that needs to be tested and especially the functionality within the release and for regression testing.

Product Manager

Issue – Quick Decisions About Releases: Rash, quick decisions are made about when a release should go out. This confuses the Product Manager and impairs her ability make a good solid decision. A well-groomed plan and process should be developed in deciding how and what should be released. Also, to assist with this problem, a well-documented backlog of issues should be managed to understand what is being impacted, and how and when it should be resolved.

Positive Change – Scrum Process: In the past analysis, when the company was adopting the waterfall process, each department was divided. Now with agile all departments are unified as one team and working together to achieve the same goals.

Issue (reiterated) – Professional Trainer for Customers: The company needs to hire and implement specified trainers to assist the customers and understand how and what they are doing. This will ensure the proper use of the software. Having a professional trainer to answer questions in a knowledgeable way will help relieve pressure on Sales, Product Management, Support and even QA and Testing.

Software Developers

Positive Change – Scrum Process: The new agile process has made development cycles shorter and found problems faster. Within the scrums, the teams and individuals meet more often which allows them to collaborate and to know what is getting done. User stories are being created which now helps the Software Developers understand what to develop from a functional perspective, rather than reading long requirement documents. In the past, the development department had one week to review requirements but now with scrums this process is more collaborative. Questions can be easily asked and answered directly by the source. This creates a faster, effective, and efficient turnaround.

Issue – Communication is needed with QA: Better communication to and from development is needed. Time should be set aside to discuss development and QA and testing needs. Although scrums are being done, they are not completely resolving all issues and questions between development and QA and Testing Departments. Often, these efforts need to be pushed without management intervention.

Issue – Defect Management and Control: Not all defects are documented nor are they indicated if they are replicated from prior issues. Better control of issues created and to be prevention needs to be a focus in the company's future.

Positive Change – Communication: The communication has been better with agile; more changes within development which with scrum then opens the communication channels with many departments all at the same time.

Positive Change – Quality: Automation testing is now used for repetitious, mundane tests. This was not used in the past, but now it creates a better quality outcome for the product.

Issue (reiterated) – Internal Training and References: Training is needed for developers to ensure that they are using the right coding standards. High-level reviews are being done, however the developers are not experienced enough to ensure that they are creating the right structural code for the projects. Management does not seem to spend the right amount of time with Development, and at the same time puts too much trust in Development to resolve coding developmental practices.

Issue – Too Much Trust in the Developers: Development has indicated that too much trust is given to them, when they are not management nor have the right amount of experience to ensure a coding change, or when and how to perform a release. They are often left wondering what to do next. They would like management to take control on decisions – both detailed and high-level.

Technical Writer

Positive Change – Scrum Process: Scrums assisted with communication and allowed great dialog that has not been achieved before with QA and software testers. The knowledge achieved within the scrum meetings have assisted the technical writer's knowledge about the teams, products, and process. Now with agile they can be working in parallel with other and know what to document for current and future needs. In the past, this was not well-planned. The technical writer indicated, "quality has been increased with using agile and no one wants to go back to the old waterfall process."

Positive Change – Communication: The technical writer is aware of the release schedules of when and what is being released. In this past, this communication dialogue would have never happened unless the technical writer went searching or asked the right individual.

CONCLUSIONS

While assessing the company's software quality during the one-day onsite follow-up study at Notify Technology Corporation, the Researcher found three aspects of positive change from the last analysis that was consistent among each of the interview areas: 1. Scrum Process; 2. Communication; and 3. Quality. The Scrum Manager also indicated a fourth positive change, which was Culture. Based on the implementation of a Scrum Process within the organization, departments were positively impacted and created positive changes, especially in areas where issues were noted in the previous analysis from 2011. This most recent analysis finds that communication was improved, which resulted in increased quality. As stated by the Scrum Manager (who was the prior QA Manager), the increased quality and positive work environment carried over to improve the company's overall culture.

To recap in the 2011 study, eight areas needed to be changed and adjusted. The eight areas are: QA Training; Customer Problem Escalation Process; Requirements Engineer; Communication Improvements; Bug Closing; Agile Software Development Process; The Creation of Publications Department; and Director of Product Development.

Based on the next steps from this follow-up case study, here are some additional changes the company should make:

Professional Scrum Training

Non-Bias Management: Now the VP is in charge of QA and Development.

Requirement Documentation: This was identified as a need by the first study and still is a major need. It seems that the company is focusing its energy on implementing a requirement management tool called Rally Software rather than setting forth the right people and processes in this area.

Permanent Trainers for Customers: Customers must be more focused and knowledgeable on how to use the end software development product. Proper training and delivery needs to be given to the customers, rather than relying on the Sales and Technical Support teams. Focusing on better training and delivery will in turn create increased quality and customer satisfaction, thus allowing Sales and Technical Support to focus on their jobs.

Development Direction: Developers need to know if their work is done well rather than the new VP having full faith and trust in what they are doing. Developers have indicated, "I cannot believe we are making our own decisions without follow-up or direction." It seems that a software architect may need to be hired to ensure proper development structure and technique if the VP cannot assist and understand the development impacts for positive growth and maturity.

Governance Metrics: Currently QA is looking at "how many" but not at "how and why." QA needs to understand and make suggestions for why certain defects exist and how to resolve them rather than having "x-number" of defects in "x-severity" open and then closed by "x-person." An assurance of consistent quality and maturity needs to exist if the organization is to grow.

QA Automation: It seems that the company laid off many QA testers since the time of the previous analysis, and the workforce power has been diminished. There needs to be a reinvestment in test automation so as to create better efficiency within the SDLC.

Experts: When talking to the developers and quality analysts, the researcher realized that they are looking for direction. However, management is looking at these same people for direction. There aren't any strong subject-matter experts in the lower ranks, since the old QA Manager was promoted to Scrum Manager and the Development Manager as now VP of Software Development and QA. The primary researcher believes that leaders and experts within Development, QA, and Business Analytics should be hired. By doing so, the company will be able to become a leader within its industry.

Now that the company has been bought by Globo Plc., Notify Technology Corporation will continue to grow as a market leader in creating mobility based enterprise software systems. According to the second author of this paper, and now former CTO of Notify Technology Corporation, “if the suggestion of agile was not done from the first study in 2011 the company would have failed months ago.” Globo Plc. saw the value of Notify Technology’s products and processes. The value of a strong quality process – and particularly by the implementation of Agile -- Globo Plc. decided to keep the company operating as normal, allowing it to grow stronger and become an integral part of the larger organization.

REFERENCES

1. Cao, L., & Balasubramaniam, R. (2007). Agile Software Development: Ad Hoc Practices or Sound Principles? IEEE March/April 2007. (pp. 41.).
2. Condon, S. (2014). Retrieved March 15, 2014 from CBS News:
<http://www.cbsnews.com/news/firm-fixing-obamacare-website-has-history-of-problems/>
3. Dyba, T., & Dingsoyr, T., (2009). What Do We Know about Agile Software Development? IEEE September/October 2009. (pp. 6.).
4. Feldman, S. (2006). Quality Assurance: Much More than Testing. Retrieved February 10, 2008, from Association for Computing Machinery:
<http://www.acmqueue.org/modules.php?pa=showpage&pid=276&name=Content>
5. Globo Plc. (2014). Retrieved March 15, 2014 from Notify Technology:
<http://www.notifycorp.com/announcements/notify-technologies-inc-business-and-assets-acquired-by-globo-plc>
6. Notify Technology Corporation. (2014). Retrieved January 16, 2014 from Notify Technology:
<http://www.notifycorp.com/about/index.htm>
7. Papows, J. (2011). *Glitch - The Hidden Impact of Faulty Software*. (pp.101) New Jersey: Prentice Hall PTR.
8. Scarpino, J.J. & Chicone, R. (2011). 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
Issues in Information Systems Volume XII, No. 1, pp. 7-15, 2011
9. Scarpino, J.J. & Kovacs, P.J. (2008). An Analysis Of A Software Quality Assurance Tool’s Implementation: A Case Study’ *Issues in Information Systems* Volume IX, No. 2, 2008
10. Schulmeyer, G., & McManus, J.I. (1999). *Guide to Software Quality Assurance*. (pp.7) New Jersey: Prentice Hall PTR.
11. Sennikovski, J. (2010). Transitioning From Waterfall to Agile: The Test Team Perspective. *Software Quality Professional*; Jun 2010; 12, 3. (pp. 4.).
12. Turk, D., France, R., & Rumpe, B. (2005). Assumptions Underlying Agile Software-Development Processes
Journal of Database Management Oct-Dec 2005; 16, 4 (pp. 62).
13. Van Genuchten, M., & Hatton, L. (2010). Software: What’s In It and What’s It In? (pp. 14). IEEE January/February 2010.