Agile Adoption Case Study, Pains, Challenges & Benefits

Asmaa Anwar
Software Engineering Competence Center (SECC)
Giza, Egypt
ahanwar@itida.gov.eg

Amr A. Kamel
CS Dept., Cairo University.
Software Engineering Competence Center (SECC)
Giza, Egypt
aakamel@itida.gov.eg

Elmohanned Ahmed
Link Development
Cairo, Egypt
elmohanned@gmail.com

ABSTRACT
An organization with an established process, assessed at CMMI maturity Level 3 more than 10 years ago, realized process corrosion symptoms. Over the years, changes in business conditions resulted in more and more process tailoring rules to adhere to the new needs, while maintaining the process maturity level. The increase in tailoring rules has made the process definition hard to understand with inconclusive tailoring results in some cases. This lead to major changes in the organization's process improvement strategy.

To build a sound process and avoid previous improvements pitfalls, the process improvement team decided to research the applicability of process agility before adopting it. The question we tried to answer was "Can agile processes improve the organization's performance indicators while maintaining the process maturity level?" We used action research, a practical and widely used research methodology, to answer our question. We selected action research to engage process actors in the improvement and/or refinement process.

In this paper, we report on our improvement efforts during the first year of deployment. This paper focuses on issues, pains and challenges we faced during the first cycle of redefining and deploying the processes to achieve the agility goals. Results from the first deployment cycle, were promising. Among other improvements, process agility reduced the rate of rework from 29% to 13%.

CCS Concepts
• Software and its engineering →Agile software development

Keywords
Agile software development; practical experience; action research

1. INTRODUCTION
Adapting the process to adhere for new business objective is more of a routine task for an organization deploying a well-established and well-maintained process, assessed at CMMI maturity level3. Among others, reducing development cost was set as a strategic objective for a medium size, software development organization located in Egypt. Following the typical improvement tasks to adhere to the newly set objectives the quality assurance team started by reviewing the development processes. However, careful review for process measurements and quality assurance activities revealed an improvement paradox. Over the years, successful process improvement efforts, has caused the overall development process to suffer from corrosion symptoms. At one hand, many fine tunings for the process managed to meet its objectives by providing tailoring rules to cope with required process dynamicity. On the other hand, the amount of added rules made the process harder to understand, difficult to tailor efficiently, adding inherent rigidity to the process. This accumulation of tailoring rules has caused what call "overall process corrosion" – a process that works but very hard to evolve.

As a result, senior management decided to create a dedicated Software Process Improvement Group (SPIG). Headed by the quality assurance manager, the SPIG was assembled as a subunit within the quality assurance department. The newly created subunit hired two full-time process improvement engineers. The sole mandate for the SPIG was to align the development processes with the new strategic objectives through a dynamic and flexible development process. To assure process dynamicity, the SPIG refrained from micro improvement actions and addressed the goals through a holistic view that addresses the development activities as well as the improvement efforts.

In section 2 we will discuss why we decided to adopt agile development methods, section 3 gives an overview on our process improvement approach. Section 4, is dedicated for explaining the challenges we faced during deployment along with our strategies and methods to overcome these challenges. Section 5, outlines the process resulting from our improvement experience. The results are discussed in section 6 and section 7 contains our conclusions and future directions.

2. WHY AGILE?
Breaking the "Development Cost" metric into its contributing factors, two sub-goals of "Reducing Internal Testing effort" & "Reducing User Acceptance Testing (UAT) effort" were identified as the major contributing factors. Further analysis showed longer than expected UAT periods is the root cause for a considerable percentage of projects scheduling and budget overruns. Hence, the SPIG decided to focus on reducing testing effort and UAT periods in its first improvement cycle.

Published reports in the area, from practitioners [1], [2], [3] and research communities [4], [5] has presented agile project management techniques has solved similar problems in different contexts. Furthermore, surveys in the area [3], reported that the ability to meet client needs and the delivery of quality software products on time are significant benefits of agile project management techniques. This was the main driving force to base our process improvement initiative on agility principles.

The SPIG administered a series of brain storming sessions with different process stakeholders to introduce the agility principles,

AMECSE '16, May 28-29, 2016, Cairo, Egypt

DOI: http://dx.doi.org/10.1145/2944165.2944175

the published results, and to reflect them upon the company internals. As a conclusion, the SPIG was convinced that the required alignment between the development process and the company strategic goals are ought to be achieved if an Agile Mindset is well developed within the developers. Upon presenting brainstorming results and the reflection on our company specifics to management, the concept was approved. Management declared an initiative named "Go Agile" for that purpose. Accordingly, transition to agile was assigned as the mission of the Process Improvement subunit.

3. OUR APPROACH

To achieve the holistic we wanted, we decided to follow an established scientific methodology to reach our goal. In addition to improving the process, we wanted to build the capability within our team to tackle the challenges that lie ahead. The target is to follow a method that is incremental, practical and assures the involvement of all stakeholders. For that purpose, we decided to base our improvement approach on action research [6]. The approach has four distinct phases: analysis, fact-finding, implementation and evaluation. The idea is to follow a structured methodology to incrementally, adopt the agile techniques and systematically build the knowledge within the organization. The underlying objective of including projects incrementally is to allow reflection upon each project experience and refining the process for the next deployment cycle.

In the following subsections, we will briefly elaborate on the four phases of the action research.

3.1 Analysis

During the analysis phase, the research team studies the research mandate. Upon analyzing the mandate, the team should propose a concept for improvement to the organization main stakeholders. If the concept is approved, the investigation phase starts in view of the proposed concept.

3.2 Conceptualization & Planning

During the conceptualization & planning phase, the team surveys related literature, and spends time studying existing processes, scoping and defining challenges in detail. The team also decides on the evaluation methods that they would use during evaluation phase. Evaluation methods focused on validating success or failure of suggested changes as well as understanding how and why the changes worked/failed. By the end of this phase, the team develops an improvement plan and acquires plan approval from management. The plan should also include justifications for different choices made.

3.3 Implementation

Once the plan is presented, the management either approves the suggested plan or requests more detailed investigations. A proof of concept may be requested or clarification for some specific points. If further investigation is required, the team holds focus group meeting and/or further researches existing literature to dig deeper and provide answers to management concerns. Upon coming to a consensus on the plan, the team facilitates the implementation of the new processes.

3.4 Evaluation

During the evaluation phase, the SPIG evaluates the implemented changes. The evaluation should include all the quantitative and qualitative measures identified within the investigation & planning phase. The evaluation phase identifies the success or failure of planned changes. The team also reflects on the evaluation results to understand the reasons that lead to success or failure the changes.

4. CHALLENGES

To establish an agile mindset within the organization, the concept for improvement approved by management, we held a series of brainstorming sessions to identify the foreseen challenges. Identified challenges were classified as urgent and non-urgent based on the organization specifics. The team decided to deal with all urgent challenges in this improvement cycle. Non-urgent challenges were left to the next improvement cycles for further data collection and analysis.

As report by Vijayasarathy et al. [3], indicated that organizational resistance, management apathy, inadequate training, lack of peer support, lack of formal guidelines, minimal rewards for using agile techniques and increased risk of project failure are the main reasons for challenges in the adoption and use of agile processes and methods. In addition, we identified two extra urgent reasons for challenges during the brainstorming sessions. Existing organization structure presented one challenge; specifically the structure of the cross functional teams. The organizational culture presented another challenge manifested in concerns raised by the testing team.

Since management was the originator of this improvement mandate, the team envisioned management apathy as non-urgent. Moreover, the SPIG deferred dealing with "lack of peer support" to latter improvement cycles. The underlying idea was "generating a success story will deep this concern as non-existing.

Careful analysis of the above-mentioned reasons reveals that they are interrelated and could be addressed through common strategies. For example, both the organizational resistance and lack of peer support could be attributed to either fear of leaving the comfort zone or lack of knowledge. Furthermore, a suitable training program handles lack of knowledge and inadequate training together. The fear of leaving the comfort zone, minimal rewards for using agile techniques could be addressed by building the right agile culture.

In the following subsections, we will discuss the strategies we used to control the above-mentioned challenges.

4.1 Building the Agile Culture

Starting with an enthusiasts team, provide the right training and exhibit a success story was our steps to build the right culture. The delivery unit responsible for projects located in Egypt was the right selection for the enthusiast team. The software delivery manager for that unit was actually pushing for going agile. New to the position, with thorough knowledge of agile project management practices and previous experience in similar agile mindset-change projects. Starting with a believer who wants to make it happen is a major driving force for success. Despite the challenges and the pitfalls at implementation start, the unit manager believed in agile values, and was determinant to get them realized. It was obvious that a person passionate about the initiative, in management position is a major driver for success.

On the other hand, almost no one from the development team was buying in the idea. The team consisting of almost 50 engineers (technical leads, developers, and Testers) had only 5 who participated in a semi-agile project. This nucleus consisting of a Project Lead, a Technical lead, 2 developers and one tester, felt the difference and they were willing for another trial with more structured and well-defined process. Those were not enough to start the big-bang implementation with, but they were the spark among others who were either fighting the idea or just living another day, with the feeling of "Nothing will work". Training was essential as a complementary step to build the right culture.

Meanwhile, management put the agile related activities "on the spotlight" to show its importance to the organization strategic objectives. Creating a project, with a catchy title "Go Agile" is an example of these glorification activities.

4.2 Training

In addition to being a reason for challenges of its own, lack of agile knowledge was the major contributing reason for organizational resistance. Few practitioners confused agile project management with chaotic or no management – a common misconception that confuses agile project management with fragile project management or no management at all. Others based their opposition on "we are implementing the right thing" argument. Due to tight schedule, we selected classroom training as the proper training option in our case. Members of the delivery team, process improvement team went through a classroom style agile training. The training covered,

- Agile core concepts & principles
- Scrum overview, main roles
- Release & Sprint planning
- Agile Tracking & Retrospectives
- User story writing

Although most of the implementation was on mooch setup, it managed to provide real flavor of an agile project through dividing the topics into releases and iteration, tracking the velocity and the progress on burn-down charts and conducting retrospectives. We noticed impact of training almost instantaneously; most of the attendees were very thrilled for applying what they learnt on their projects. On a hindsight, discussions during the class, the quality of the instructor contributed to this enthusiasm.

4.3 Project Management Adjustments

During early deployment of a new process, risk of project failure, is always a concern. In our case, this risk manifested itself in different ways: (1) Risk of rejecting the product by the customer, (2) risk of estimating the required effort wrongly and (3) the lack of experience with project tracking tools and techniques, namely the stand-up meetings & burn-down charts.

Our new process deemed requirement sign-off with customer and between developers and testers unnecessary. Although the sign-off mechanism did not guarantee neither the mutual understanding of requirements between developers and testers nor the requirements changes from the customers, it provided some sense of security to both managers and testers – what if the customer refused to accept the product was a major concern to project managers.

To deal with this sense of insecurity, we replaced requirements sign-offs with regular "Client Demos" scheduled at the last day of each sprint. During a Client demo, the project manager and the technical lead present latest product release to customer.

The other manifestation were directly attributed to lack of experience from the project team and lack of historical data to build a reliable estimation database. Support from a highly motivated Technical Lead diminished the impact of lack of experience, and we overcame the lack of historical data by using "commitment based planning" for initial sprints; later sprints followed the same planning process, except for having recorded team velocity to support the planning process. The first project started by having requirements for the first release set as "Release Backlog".

We had initial rounds of "Story Writing" and "Release Planning" sessions where the whole project team (technical lead, developers and testers) were invited. During a typical planning session, the technical lead acted as the product owner in this phase and explained selected user stories for the sprint. The team discussed each user story to confirm their mutual understanding and then estimated the story size using "Planning Poker" [7]. For each story, testers provided the "acceptance criteria".

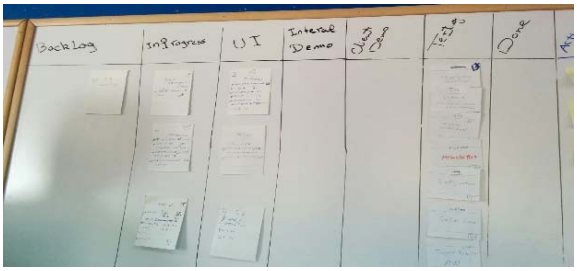

Figure 1 Sample Task Board

Finally, while adopting the typical management tools in agile setup, e.g. burn down charts and task boards, the team decided to use physical management tools to manage the development process (see for example the sample task board in Figure 1 Sample Task Board). On top of its role as management tools, they supported knowledge dissemination among project teams.

4.4 Addressing Organization Structure Limitations

Cross-functional teams (teams of specialists serving all projects within the organization) are one of the main building blocks of the organization. For example, the organization has a team of eight System Analysists (SA) serving analysis function across the whole organization. This team is serving three delivery units with many projects. This organizational setup presented a reason for challenge in the process. Traditionally, only members of the SA team get face-to-face contact with the customer. Which made them best candidates of the Product Owner role (PO) in the new process set-up. The PO has to be available full-time during the whole project life span. Hence, assigning a dedicated member of the SA team to each project means that the company can work on a maximum of 8 projects simultaneously. Historical data for our target delivery unit indicated that the peak implementation load is four concurrent projects, implying that 50% of the analysis team capacity might be consumed within one delivery unit; which will negatively affects work in other delivery units.

The SPIG presented two obvious solutions; either, increase the analysis team capacity, or assign the role to someone else. Senior management did not accept either options. At one hand, increasing team capacity is not justifiable to management in view of "reduce development cost" as strategic option. At the other hand, experience in interfacing with customers possessed by analysts is critical for project success. Alternatively, we overcame this challenge by spreading the PO role. The Project Manager (PM), the Technical lead (TL) and the system analyst (SA) held the PO role collectively. This implied that the PM and TL were invited to all requirements gathering & validation sessions. Despite the extra cost added by this setup, the projects gained some benefits. The direct contact between the TL and the PM with the customer raised their awareness regarding project context, which paid back during UAT phases.

4.5 Raising Confidence within the Testing Team

Traditionally, "Requirements review" was the main vehicle to establish mutual understanding between developers and testers. Both teams expressed their mutual understanding in internal requirements sign-off. The sign-off is the result of offline reviews

held by testers followed by walkthrough sessions conducted with System Analyst and Technical Lead. This process was cancelled due to testers' involvement in story writing and release planning. This change to the testing process raised technical concerns from the testing team.

Despite their early involvement in the development cycle, the testing team felt uncomfortable with the new process. User stories replaced the analysis document. From their perspective, the format and content of the user stories did not seem a technically sound alternative to the analysis document – a formal document they usually use to lead all test case generation efforts. The introduced change affected their basic professional mindset.

These concerns presented a mixture of culture and technical concerns. Our resolution strategy was to add an extra step, preferably from existing organization processes, to raise the testing team confidence with the products through an established process within the organization. The smoke test process from existing organization processes database played the desired role. It ensured a minimal quality for the product before going under test through a trusted process.

5. PROCESS DEFINITION

The SPIG decided to adopt an agile approach to define the development process. The process was defined as per-need basis. We started with some guidelines and filled in needed details upon project experiences to provide a grounded process that provides real support to our developers than a process that does not fit our organizational needs. SPIG team members played the "Agile Coach" role to support process deployment and to document the day-to-day experiences to fine tune the process at later stages. Process details were initially drawn from the training material and related books (see for example [8])

The SPIG applied the action research methodology rigorously to turn documented experiences into process knowledge. We used the Generated knowledge in process guidance, "dos and don'ts", and to steer process definition activities. In the next subsection, we will present the final process resulting from this exercise.

5.1 Development-Testing Sprints

Close interaction between testers and developers supported building trust between the two teams. Overtime, testers felt more comfortable about the quality of the product under test, hence, their initial concerns discussed in section 4.5 diminished and the reason for the smoke test disappeared. The second round of process deployment dropped the smoke test completely with no measurable impact on product quality.

In the second round of deployment, a workgroup representing the development team, the testing team and process improvement team revisited the development process. The workgroup redefined the development-testing sprints lifecycle. Typically, a user story goes through three consecutive quality gates in each sprint.

- Upon developing a user story, the developer deploys the product on the testing environment and declares the story "ready for internal demo".
- During the internal demo – the first quality gate – the developer demonstrates that user story works and shows that it meets the acceptance criteria. This demo takes place at the developer desk. The technical lead, developers, testers and the agile coach attend the demo. If the team accepts the implementation, it passes the second quality gate and the story is declared as "Ready for Client Demo".
- During the client demo – the second quality gate – the project manager and technical lead present the product to customer to gain customer acceptance. This demo takes place at the client site. Usually, Client Demos are performed at last day of each sprint.
- At the top of its iteration backlog, the following sprint includes the comments resulting from the Client Demo. The development team handles the client comments, if any, and declares the story as "done".
- Integration Testing – the third quality gate – starts at this point.

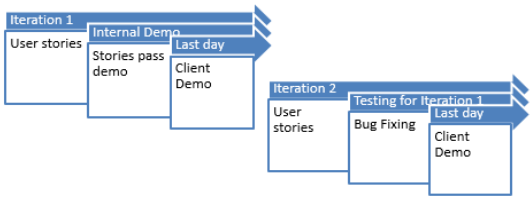

Figure 2 Development Lifecycle

As a result, a typical user story is completed in two consecutive sprints. The first sprint for development, ending with client demo, the second sprint is for testing. Hence, the development-testing activities were interlaced. Starting from the second sprint, sprint backlog will contain "Bug Fixing" tasks for stories implemented in the previous. This interlacing allowed us to dedicate the last iteration of each release for release stabilization with testing & bug fixing activities only as seen in Figure 2.

6. PILOT RESULTS

The Go Agile project resulted in a new development process based on agile concepts. Over the span of (Jan 2015 to Dec 2015), the SPIG piloted the process on five projects within one development unit. For this cycle of deployment, we chose projects that are completely developed in Egypt for Egyptian customers.

To judge the success or failure of the process improvement project, the SPIG conducted a set of qualitative and quantitative analysis to evaluate the experience from two perspectives, business and internal impact. For the business perspective, we selected a set of performance metrics that (1) are directly related to the company strategic objectives, (2) could be compared against the company baselines and (3) can show the impact quantitatively. For the internal impact perspective, team was surveyed to measure their satisfaction with the changes in the development experience.

6.1 Performance Metrics

The SPIG selected performance metrics from the organization database directly related to "reduce development cost" to judge impact on the strategic objective in focus. The selected metrics are:

- "Testing overhead" defined as the total effort consumed in testing calculated as a percentage of total effort consumed in code writing.
- "Rework" defined as the total effort consumed in fixing defects calculated as a percentage of total effort consumed in code writing.
- "Leaking-defects rate": the number of found defects that belong to the basic coverage category calculated as a percentage of the total number of found defects. – The Basic Coverage Category is a set of defects that should

have been discovered (and fixed) before the testing phase[1].

On top of these measures, we used "Customer Satisfaction", as a sanity check measure. Customer satisfaction is the average rating for a product given by the customer. Customer Satisfaction is collected by surveys where customers rate different aspects of the product on a scale from 1 to 5; with one is the lowest and 5 is the highest. Changes in Customer Satisfaction Rating represents cost-on-business for any proposed improvements, as they summarize the company performance as envisioned from the customer side.

To judge the success or failure of our process improvements, the SPIG defined success criteria for the improvement project. The success criteria required significant improvements in at least two of the three selected performance metrics. The improvement is measured against the organizational baseline calculated in 2014 given that customer satisfaction is not negatively affected.

Table 1 Performance measure results

Metric	2014 Baseline	Projects' average
Testing Overhead	34%	26.4%
Rework	29%	13%
Defects Leakage	93%	72.2%

"Customer Satisfaction" ratings for all projects was above the 2014 baseline (4.0). In fact, the average for the rating for pilot projects is 4.3 scoring a slight increase over the baseline. Furthermore, performance metrics were improved in all projects within the study showing significant enhancements in all areas directly affecting the company strategic goals. As shown in Table 1 Performance measure results, Rework decreased by 55.2% recording the highest value of improvement. This significant decrease showed itself in UAT duration reduction. The average UAT duration for the completed projects (3 projects) was 10% of the total project duration. Testing overheads and defects leakage metrics improved as well with more than 20% enhancement over the baseline.

Much of the performance improvement could be attributed to the improvement in intercommunication and level of trust between developers and testers. For example, replacing the smoke test (typically consuming 2-3 days activity) by Internal Demo (a 2-4 hours meeting) reduced the total testing effort and the overall project time.

6.2 Team Satisfaction

To check the internal impact of our process improvement project we surveyed all team members working in the pilot to measure their satisfaction level with aspects of the new process. We centered the survey questions around comparisons between new processes and established processes. Team members rated on a scale from one to five how they like certain aspects of the new process as compared to the established process. The average of all questions for all team members is the team satisfaction rate. For the projects implemented so far, the team satisfaction rate is 4.0.

The SPIG also collected testimonials about the new process. Testimonials pointed out that more organized work environment, better team collaboration and aligned project vision are the major benefits of the agile process. Upon elaborations with team members they attributed these enhancements to the process organization that led to better understanding of client needs and vision. Furthermore, the testimonials also added increase in product quality; however, company metrics did not support this claim. Quality related metrics stayed within company baseline averages. However, rework measure suggests that the quality of the developed code (rather than the final product) has increased.

6.3 Reflections

6.3.1 Testing time

In a typical development environment, testing takes at least two cycles (if not more) one cycle before delivering to the customer and another cycle after considering customer feedback to gain customer acceptance. The deep involvement of the testing team in user story development and the acceptance criteria developed for each story during story writing, answered the question "How are we going to test this?" early on. This raised the developers' awareness about the testing issues and concerns. Consequently, developed user stories suffered from less and less technical issues related to the testing process. Finally, testing activities before gaining initial customer acceptance deemed unnecessary, hence, replaced by the internal demo.

This process adopts the main principle of the TDD (Test Driven Development) [9] development method, yet in a less formal, easy to implement way. Despite the added cost of having the testers involved in user story writing, this process decision paid off by shrinking the total project duration. In fact, this particular process choice was a main contributor to the reduction in testing overheads.

6.3.2 Management of Meetings

Meetings used to have bad reputations in our company; meetings are usually seen as hectic, lengthy process, which add little value to the development. Coming from this background, team members strongly resisted the daily standup meetings. Initially, the SPIG pushed on keeping the meetings in focus and forced the meeting to close on schedule. Over time, project teams gathered for standup meetings on time and update status even if the technical lead and project manager are not available. Eventually, the purpose of the standup meeting expanded from status reporting (keeping team members up-to-date) to include self-organization issues.

Although nothing new regarding the impact of good management of team meetings, we found out that the quality of the standup meeting is a prime indicator for the quality of whole process. Furthermore, we used commitment to attending these meetings as the indicator for culture change within project teams.

6.3.3 Physical Management Tools

On top of managing projects, the physical management tools served other intended purposes, publicity within and outside project team. Within the team, making key project aspects (tasks, status and schedule) publicly available ensured the transparency of project scope and deadlines. This facilitated the creation of common goals within the teams. Outside the team, the availability of theses information, created the required hype and curiosity about the agile methodology within the entire company.

Unfortunately, keeping physical tools up-to-date is not an easy task. For the purpose of load reduction, the Agile Coach, who attends the daily standup meetings, took over the update responsibility. We wanted project team members to appreciate the importance of these tools with no added overheads.

[1] The organization process presents clear criteria to identify defects that fall within the Basic Coverage Category.

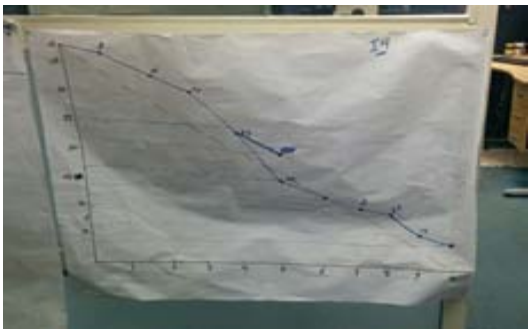

Figure 3 Sample Burn-Down Chart

After some experience with these tools and process-setup, other benefits showed up. The regular update of the burn-down chart (see Figure 3) by the agile coach was a major motivating factor for the project teams; team members were racing to finish tasks before update time. Overtime, charts location turned to "meeting points" for project team members. They used to discuss issues related to the project while checking the project status and/or progress – as a result, team intercommunication was greatly improved.

7. CONCLUSIONS

In summary, this paper reports on a major process improvement project in a medium size development company. The project goal is to move the company into agile development methods systematically. We followed action research methodology to build the process. Replacing the "textbook sprints" by "Development-Testing" interlaces reduced the total development time and provided better workload distribution between developers and testers.

While working on our process we learnt that, the first defined process is a through away, do not spend much time perfecting it. Addressing human based challenges are as important as addressing technical challenges, if not more important. Unfortunately, these challenges depends on the organization specifics. However, organization structure and trust between different project roles and teams (who will be held responsible) are the main origins for these challenges.

By the end of 2015, senior management reviewed the results. Approved an adoption plan for the process across the company. The new adoption cycle, ending by the second quarter of 2016, will include projects from two other delivery units. New improvements will consider challenges within the process like managing epic scenarios that result from complicated business scenarios and working with offshore clients.

8. REFERENCES

[1] J. Highsmith and A. Cockburn. 2001. Agile Software Development: The Business of Innovation. *Computer,* vol. 34, no. 9 (2001), 120-122.

[2] A. Begel and N. Nagappan. 2007. Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study. in *First International Symposium on Empirical Software Engineering and Measurement*, 2007.

[3] L. R. Vijayasarathy and D. Turk. 2008. Agile Software Development: a survey of early adopters. *Journal of Information Technology Management,* vol. XIX number 2 (2008), 1-8.

[4] P. Abrahamsson, K. Conboy and X. Wang. 2009. 'Lots Done, More to Do': The Current State of Agile Systems Development Research. *European Journal of Information Systems.* vol. 18 (2009), 281-284.

[5] T. Dingsoyr, S. Nerur, V. Balijepally and N. B. Moe. 2012.A Decade of Agile Methodologies: Towards Explaining Agile Software Development. *The Journal of Systems and Software,* vol. 85 (2012), 1213-1221.

[6] D. Coghlan and T. Brannick. 2010. *Doing Action Research in Your Own Organization*. Sage Publications Ltd..

[7] M. Cohn. 2005. *Agile Estimation and Planning*. Prentice Hall.

[8] M. E. Moreira. 2013. *Being Agile: Your Roadmap to Successful Adoption of Agile*. Apress.

[9] K. Beck. 2003. *Test-Driven Development by Example*. Addison Wesley.