Usage of SCRUM Practices within a Global Company

Mauricio Cristal
mauricio.cristal@gmail.com

Daniel Wildt
FACENSA, Brazil
daniel@facensa.com.br

Rafael Prikladnicki
PUCRS, Brazil
rafaelp@pucrs.br

Abstract

Global companies that experimented extensive waterfall phased plans are trying to improve their existing processes to expedite team engagement. SCRUM has become an acceptable path to follow for those companies because it comprises project management as part of its practices. SCRUM has been used with the objective of simplifying project control through simple processes, easy to update documentation and higher team iteration over exhaustive documentation. Instead of investing team effort on producing static documentation, SCRUM proposes to focus on team continuous improvement aiming to add value to business processes. The purpose of this industry report is to describe two projects that experimented SCRUM practices within a globally distributed company. This company has development centers across North America, South America and Asia. This report covers challenges faced by the project teams, strengths and practical recommendations of using SCRUM in a globally distributed environment.

1. Introduction

Distributed software development has been increasing during the past years in technological global market [1, 2]. Various companies are outsourcing to third party consultant companies on emergent countries, while others decided to open captive development centers with full-time employees on emergent countries in order to develop internal Information Technology assets (Internal Offshoring) [3]. Organizations aim at the development of large projects on offshore centers (1+ year duration). Reason for this decision is that long duration projects would be more suitable to absorb offshore centers learning curve about domain knowledge [4].

Usually, an initial strategy defines that offshore centers would responsible only for development effort [4]. This scenario supports an adoption of waterfall lifecycle as standard lifecycle. Onshore teams are responsible for planning and testing efforts while offshore teams are primarily assigned to developing tasks. While captive offshore development centers are increasing their business domain background they continue to follow waterfall life cycle, and its exhaustive documentation associated to waterfall lifecycle can impact team's productivity negatively.

Team members that have an acceptable business background start to struggle on processes and controls that do not make sense to their daily activity. Thus is not acceptable to invest so much effort on documenting and creating controls as used on the initial strategy of the offshore centers.

In this context, this report describes the experience on piloting agile practices (more specifically SCRUM practices) in two projects within a global company that have waterfall as standard life cycle. We present challenges faced by the project teams, as well as strengths and practical recommendations

This paper has the following structure: Section 2 presents agile and SCRUM concepts. Section 3 contextualizes the organization and describes the experience with the two projects. Section 4 covers a discussion with strengths and practical recommendations on using SCRUM on global companies. Section 5 concludes the paper.

2. Agile software development

Effective communication, focus on people, listen the customers, and always looking at improvement opportunities are some principles that can address what being Agile is about [5]. When applying agile methodologies, teams look for an agreement to deliver working and valuable software to its final users in short development cycles, also known as a time box.

SCRUM [7] and eXtreme Programming (XP) are examples of Agile Methodologies teams can use [6]. While XP focus on programming practices like Test Driven Development, refactoring and simple design, SCRUM is focused on planning [6]. Since both are based on the Agile Manifesto [5], they look for motivated individuals, and ways to make teams evolve and grow on each iteration cycle.

Teams focusing on SCRUM follow some practices that will lead them for a definition regarding what will

978-0-7695-3280-6/08 $25.00 © 2008 IEEE
DOI 10.1109/ICGSE.2008.34

IEEE computer society

be implemented within a release and, after proper prioritization, what will be implemented as part of an iteration. In a distributed team, the implementation of an agile methodology is not a simple task. It is important for the team to change its mindset and prepare for commitments and a real new enterprise culture [8].

The mechanics of SCRUM expects a team to build a product backlog, a place where one can see all requirements pending for a project, sized based on complexity, days or some other unit of measure the teams decide. Inside a product backlog you have a simple sentence for each requirement, something that will be used by the team to start discussions and details of what is needed to be implemented by the team.

The prioritization process happens with the help of the project team. SCRUM defines simple roles [7]:

- The product owner: responsible to be the voice of business inside a SCRUM project. May be also seen as the business partner role.
- SCRUM Team: formed by its developers, testers and other roles you can find
- SCRUM Master: responsible to keep the team focused on the practices and values that are needed to be applied inside the project and also responsible to help the team whenever they face some problem during the process.

After the prioritization, the team has a decision about what will be part of a release. That is the release planning meeting. Based on that, other meeting is planned, where the team details features that need to be implemented as part of a time box. Within that time box the team implements all tasks expected for the project, design, coding, testing, inspection and validation for instance. This is the Sprint. The duration of a sprint can change, from one week to four weeks.

The sprint planning meeting occurs in the beginning of each time box. The number of sprints in a release is defined by the team. It may have two or even ten sprints. It is all based on the goals of the release. If the focus remains the same, one can still continue to have more sprints. If the project focus changes, it is better to close the game, finish the release and start a new one, with new goals and expectations.

During the sprint, the team works with all they have for one specific feature. It is needed to get one feature "done" to start another. Done means a feature ready to go to production, fully coded, tested and with quality. The design is another important topic. Design needs to be done during the sprint, and a team can use artifacts they find useful, like UML modeling, or even focusing on acceptance criteria and testable tasks, proved by using unit or functional tests. No matter what is the decision of the team regarding a way to document and formalize what will be done, it is important to question if what is being done is adding value to the process or not. Waste elimination and continuous improvement is always on the top practices when applying agile principles [9].

But the usage of agile practices is still maturing in the IT industry. Looking at distributed teams, there is the need for more attention of the whole team in the communication process. Different approaches are needed in this case, based on how teams are organized. Teams can be isolated, not working in a cross functional environment, distributed across regions, or fully distributed [10]. This experience report will focus on the distributed SCRUM of SCRUMs approach, where you have different SCRUM teams across regions and short meetings to integrate these different regions [18]. We can find several studies in the distributed development literature related to agile development in distributed environments [11, 12, 13, 14, 15, 16, 17, 18]. While most of those studies are being conducted by research groups [11, 12, 13, 14, 17] we still have little experience shared by the software industry [15, 16, 18].

3. Experience report

The experience described in this paper was lived within a global organization that maintains software development centers in Americas and Asia. Technical teams work only for captive offshore development centers. This company developed a set of global standard processes following three market standards:

- Capability Maturity Model (CMM) ®
- PM Body of Knowledge (PMBoK) ®
- Microsoft Solutions Framework (MSF) ®

In order to disseminate software development culture and processes globally, this organization decided to pursue Capability Maturity Model (CMM) level of maturity assessment. Available set of processes were only based on waterfall life cycle. Project team members follow available processes and phases, regardless project size and complexity.

According to the organization strategy and a set of processes available at that time, documentation is generated onshore. Software product is developed offshore and tested back again onshore. In the past, distributed location justified extensive documentation while offshore centers were dealing with their domain learning curve acquiring business knowledge.

Nowadays this scenario has changed. Offshore centers have 5+ years of background on application domain knowledge. They are working on large, medium and small projects. In addition responsibilities have increased from "only development center" to a

center responsible for planning projects, developing, testing and requirements engineering.

According to set of defined processes on this organization project life cycle should follow extensive waterfall phases generating a significant amount of required documentation. Lessons learned showed that for larger projects (1+ year duration) standard practices were suitable. On the other hand, for medium and small projects (up to 9 months duration) standard practices were adding too much of complex and bureaucracy to team members. In addition business was not satisfied with results achieved by the end of project as also captured on lessons learned sessions.

For medium and large projects documentation produced usually were outdated by the end of project because business most of times were maturing their business requirements along the project. Due to this dynamic and changing environment some issues were listed as possible causes:

1. Powerful project managers were putting processes up front of business interests causing frustration on business expectation.
2. Weak project managers were accepting so many change requests flooding team work by paper work in order to keep documentation records up to date. Some projects had decided to have two project managers. One to track project status and another one only to maintain documentation. This practice was causing team members frustration because they were moved out from technical tasks.

Projects lessons learned sessions also showed that benefit of documentation was low in comparison to effort required to generate and maintain it. Often developers with new assignments were contacted to resolve issues on past delivered projects. Application support teams instead of using existent documentation were reaching developers to resolve production issues.

Those facts demonstrated that processes should be adapted to medium-small projects on this company. Organization decided to innovate on medium-small projects life cycle. It was clear that waterfall was not working for short-term projects. Agile methods were then selected to be run on two pilot projects.

An assumption was that agile methods would bring more interaction between business partners and technical team. Moreover, unstable requirements would be gathered on iterative way along the project as business partners were expecting. The two projects will be referenced in this paper as Project A and Project B.

Project A is a finance domain project related to global taxes control on finance business processes. This project had eight team members collocated in North and South America.

Project B is a logistics domain project related to outbound and fulfillment processes. This project has ten team members collocated in North and South America and Asia as well. Table 1 depicts both pilot projects properties in summary:

Table 1. Projects analyzed in this paper

	Project A		**Project B**	
Business Domain	Finance		Logistics	
Roles and locations	Role	Loc.	Role	*Loc.*
	1 Product Owner	*US*	*2 Product Owners*	*CHI*
	6 dev/testers	*BR*	*3 dev/testers*	*US*
	1 SCRUM master	*US*	*6 dev/tester*	*BR*
			1 SCRUM master	*US*
Team size	**8 team members**		**12 team members**	

Locations: United States (US), Brazil (BR), China (CHI)

Both pilots faced challenges while adapting existing organizational processes to agile practices, as table 2 summarizes:

Table 2. Challenges identified

Challenge	**Project**	**Phase**
Lack of formal document for requirements	A	Planning
Inadequate team structure	A, B	Planning
Communication issues	A, B	Developing
Management not used to agile practices	A, B	Closing

Lack of formal requirements can be described as the barrier of project team had in terms of capturing business requirements in an easy and effective way. Project B had developed a software requirements document but experimented difficulties on having it signed off by business partners that were located in Asia. This fact caused lack of commitment from business to agreed requirements during project development. Project A had no formal software requirements document. Requirements were gathered on exhaustive meetings (8 hours duration) and documented in minutes. Every time a backlog item required clarification, these minutes were researched impacting team members productivity negatively.

Inadequate team structure can be captured as inappropriate decisions in terms of team organization and task assignment. For instance Project A faced a barrier between generalist against specialist behavior on technical team. Agile practices states that every team member must collaborate as a generalist in project tasks. In this project even with performance testing training available test team avoided to execute performance testing because they were primarily functional testers. Root cause for this reaction is a common human avoidance of missing his/her knowledge silo and avoidance to accept changes.

Communication issues are a common challenge on distributed projects. Both projects experimented communication barriers especially during development phase. In project A, lack of team commitment on attending remote SCRUM meetings impacted team collaboration. SCRUM meetings needs to be attended by all team members in order to keep awareness on project updates. According to agile practices SCRUM meetings have a daily recurrence and it must be a very short duration meeting. Project A not only experimented lack of attendance but often SCRUM meetings lose focus getting into details of technical solutions debating unsuitable subjects with an inadequate attendance. That was a most acceptable reason on lessons learned session why SCRUM meetings lost importance during project development.

SCRUM management basis its practices for project schedule control on time-box practices. For instance it means that a team can define that every month a product release will be achieved. Requests that fits on one month time box are going to be delivered others are going to be released on following month and so forth according to request priority in product backlog. On both projects nor technical management nor the SCRUM master had enough political power to define an acceptable time box for projects A and B. Project A missed a strong leadership from SCRUM master during negotiations with business partners flooding team members with a long list of last time changes by the end of each sprint. On project B lack of management maturity on agile practices impacted development life cycle. Instead of cutting a sprint scope and postponing to following sprint, management decided to interrupt functional test. Both immature decisions caused team rework and overtime generating frustration and impacting software quality in terms of defects. These failures were fixed in the end of each iteration. The team was able to do a retrospective exercise and find ways to improve the current process. This can help the team to increase its maturity while using agile principles and SCRUM practices.

4. Discussion

Challenges were faced since the organization was more familiar with waterfall life cycle, but lessons learned have also captured a list of important strengths on using agile practices:

- Weekly software product builds was successfully used in project A: releasing as many builds as possible project team can eliminate wastes in terms of waiting for a whole package to be tested [10];

- Even though business partners took more time than expected to approve requirements document in project B during acceptance phase they participated and collaborated with project team anticipating tests and providing adequate support on defects tracking;

- Project A reported communication issues as one challenge faced. Although same project also described a strength that every sprint delivered not only increased team motivation but also improved collaboration and engagement. Comparing interaction and cooperation levels between technical teams and business partners on first sprint and last sprint a major improvement could be notice according to recorded project lessons.

4.1. Practical recommendations

Based on challenges and strengths faced, project teams were able to document a set of recommendations for projects within the same environment and scenario.

Use cases should be integrated with user stories: project A decided to separate use case diagram from user stories. Use cases diagrams were stored on a diagram tool and user stories were stored in a worksheet. During project development, keeping both synchronized and accessing its records was quite difficult and time consuming.

While piloting agile practices the team must document SCRUM meeting minutes in order to record agreements and decisions. SCRUM practices suggest to document when is necessary and useful. Lessons captured from project A recommend to formally document SCRUM meetings outcomes. Especially during piloting period, both technical team and business partners were not aware of processes limitations and responsibilities on agile practices. For this reason, recording decisions might be a positive practice. Moreover, team interaction in this scenario is limited to emails and conference calls. Since engineers and business partners are dislocated there is neither a "war room" nor white boards where decisions could be revisited when necessary.

SCRUM master needs to be a strong negotiator and also owns a *de-facto* political power over technical team and business partners. Since agile practices are based on team collaboration and trust, a SCRUM master needs to be a person strong enough to deal with business in terms of requirements priority and also to push team members to deliver on agreed time box. In a global scenario, this is even more complex. For this reason, the SCRUM master needs to combine not only leadership on project team and business partners, but also cope with distance, trust, and cultural diversity.

Assigning testers and developers to work together is an important future recommendation. For instance, assigning testers to peer review unit test scripts can improve unit tests coverage. This decision

can help on anticipating defects to development phase instead of finding only after promoting build to test team. In project B, developers and testers worked in silos, impacting team collaboration. Moreover, dislocation between engineers promoted an environment where each group created their own practices, impacting code quality.

Keeping valuable documentation: moving project B from a waterfall to an iterative and agile approach did not change the need for documentation. Since the team is working in a distributed environment, the documentation may even increase [18], and the team will need to adapt some of the documentation to become more effective. For this reason, one should define and document what brings value to the project and what can be useful to the distributed teams.

The use of a "global" taskboard can help on improving the productivity of global agile teams. The teams of both projects were using meeting minutes and local spreadsheets to control the activities planned for each sprint and the product backlog. This sometimes caused delays and unexpected problems. An interesting initiative would be the development of a tool to help global teams on the planning and execution of a sprint.

5. Conclusions

In this paper, we have presented the experience on the usage of SCRUM practices in two pilot projects within a global company. Agile methods are becoming more common in today's software development industry. Companies see the advantages of applying such practices, and make sense to also think on the usage of agile practices in a distributed environment.

Particularly in this experience report, the use of SCRUM was a challenge, since the company culture was not agile-oriented, and was hard to change the mind of several people for the advantages of applying SCRUM together with traditional practices already in place. The initial results were very positive, and future plans include the use of SCRUM in a more systematic way through the institutionalization of agile practices in a global level within the company.

7. Acknowledgements

Study partially supported by the Research Group on Distributed Software Development of the PDTI program (Law 8.248/91).

8. References

[1] Sengupta, B., Chandra, S., Sinha, V., "A Research Agenda for Distributed Software Development," In: 28th ICSE, pp. 731-740, Shanghai, China, 2006.

[2] Herbsleb, J. D., "Global Software Engineering: The Future of Socio-technical Coordination," 29th ICSE, 188-198, Minneapolis, USA, 2007.

[3] Prikladnicki, R., Audy, J. L. N., Damian, D., Oliveira, T. C., "Distributed Software Development: Practices and Challenges in Different Business Strategies of Offshoring and Onshoring", In the IEEE 2nd Int'l Conf on Global Software Engineering, Munich, Germany, pp. 262-271, 2007.

[4] Carmel, E., Tija, P., "Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce", UK: Cambridge, 2005.

[5] agile manifesto http://www.agilemanifesto.org.

[6] Beck, K.., "Extreme Programming Explained", Embrace Change. 2005.

[7] Schwaber, K., "Agile Project Management with Scrum". 2004.

[8] Schwaber, K., "The Enterprise and Scrum". 2007.

[9] Poppendieck, M., Poppendieck, T., "Implementing Lean Software Development: From Concept to Cash", 2005.

[10] Sutherland, J., Viktorov, A., Blount, J., Puntikov, N., "Distributed scrum: agile project management with outsourced development teams", In HICSS, 2006.

[11] Layman, L., Williams, L., Damian, D., Bures, H., "Essential communication practices for Extreme Programmin in a global software development team", *Information and Software Technology*, Elsevier, 48 (2006), pp. 781-794.

[12] Agerfalk, P. J., Fitzgerald, B., "Flexible and distributed software processes: old petunias in new bowls", CACM, 49 (10), pp. 27-34, 2006.

[13] Hazzan, O., Dubinsky, Y., "Can diversity in Global Software Development be enhanced by Agile Software Development?", Workshop on GSD for the Practitioner at ICSE, Shanghai, 2006.

[14] Ramesh, B., Cao, L., Mohan, K., Xu, P., "Can distributed software development be agile?", CACM, 49 (10), pp. 41-46, 2006.

[15] Flor, N. V., "Globally distributed software development and pair programming", CACM, 49 (10), pp. 57-58, 2006.

[16] Taylor, P. S., Greer, D., Sage, P., Coleman, G., McDaid, K., Keenan, F., "Do Agile GSD Experience Reports Help the Practitioner?", Workshop on GSD for the Practitioner at ICSE, Shanghai, 2006.

[17] Paasivaara, M., Lassenius, C., "Could Global Software Development Benefit from Agile Methods?", In: ICGSE, Florianópolis, Brazil, 2006.

[18] Fowler, M., "Using an Agile Software Process with Offshore Development", available online at www.martinfowler.com/articles/agileOffshore.html, 2008.