

SCRUM: Application Experience in a Software Development PyME in the NEA.

Walter G. Barrios

**Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura.
Universidad Nacional del Nordeste. 9 de Julio 1449. 3400. Corrientes– Argentina.**

María V. Godoy

**Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura.
Universidad Nacional del Nordeste. 9 de Julio 1449. 3400. Corrientes– Argentina.**

Mirta G. Fernández

**Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura.
Universidad Nacional del Nordeste. 9 de Julio 1449. 3400. Corrientes– Argentina.**

Sonia I. Mariño

**Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura.
Universidad Nacional del Nordeste. 9 de Julio 1449. 3400. Corrientes– Argentina.**

Fernando M. Ferreira

Facultad de Ciencias Económicas.

**Universidad Nacional del Nordeste. Av. Las Heras 727. 3500 Resistencia. Chaco.
and**

César T. Zarrabeitia

Facultad de Ciencias Económicas.

Universidad Nacional del Nordeste. Av. Las Heras 727. 3500 Resistencia. Chaco.

ABSTRACT

In order to manage projects efficiently, agile methodologies for software development tools that improve production processes have emerged. This paper discusses the adaptation and implementation of the SCRUM methodology in a software development company of the NEA (Northeast Argentina) used under a strategic management approach, redesigned for its use in a microenterprise. The challenge was to achieve an effective technological linkage (between management and systems) oriented to the innovation in simplifying and streamlining roles in implementing the methodology. In the paper, we introduce the subject theoretically and then expose the practical aspects of the case and analyze the results.

Keywords: SCRUM, Software Engineering, Case Study in NEA, University-Business, Interdisciplinary Work.

1. INTRODUCTION

Scrum

Software engineering [12] includes every aspect of the software production, from the initial stages of system specification up to maintenance after the software has been used. Within this area the methodology [8], allows to determine the tasks to execute in order to improve the effort gained by the human resources team involved.

The term SCRUM refers to a strategy. It was originally coined in rugby, and it is understood as to put back into the game a lost ball, in which the whole team cooperates and decides quickly the next step.

Scrum is an iterative, incremental [14] framework for projects and product or application development. It structures development in cycles of work called **Sprints**. These iterations are no more than one month each, and take place one after the other without pause.

As agile methodology specifically referred to IS, in 1993, Jeff Sutherland [14] applied the model SCRUM to the development of software in Easel Corporation (Enterprise that in the macro games of buying and merging would end up uniting in VMARK, then in Informix and finally in Ascential Software Corporation). In 1996 Sutherland presented, together with Ken Schwaber, the practices put

into use as a formal process to manage software development in OOPSLA 96 [11]. The experiments designed by Schwaber y Sutherland to manage software development are included in the agile models list of Agile Alliance [10].

In [13] some implementations of SCRUM are mentioned. Takeuchi y Nonaka [15] observed for the first time diverse variants of the approach of such a methodology for the development of new products with small high performance equipments in enterprises like Fuji-Xerox, Canon, Honda, NEC, Epson, Brother, 3M, Xerox and Hewlett-Packard. Coplien [3] indicated a similar approach applied to software development in Borland. Besides Sutherland [13] used a refined approach of this process to development in Smalltalk and Schwaber [11] and to production in Delphi.

SCRUM is nowadays, used by big and small companies, including Yahoo, Microsoft, Google, Lockheed Martin, Motorola, SAP, Cisco, GE, CapitalOne and the USA Federal Reserve [4].

Its basis is team work motivation and an agile reaction to change oriented to the final result. Those projects which requirements change rapidly are presented as the most appropriate ones for its application. Its main characteristics may be summarized in [1]:

- Software development is carried out through iterations, named Sprints, which last for 30 days. The result of each one of these is a feasible increase that is shown to the client.
- Meetings all along the Project; it is noteworthy that the development team meets daily and briefly, only for 15 minutes, for coordination and integration.

The practices applied by SCRUM to maintain an agile control in the project are: i) revision of iterations, ii) incremental development, iii) evolutionary development, iv) team self-organization and v) collaboration.

The key roles, artifacts and main events are summarized in Figure 1.

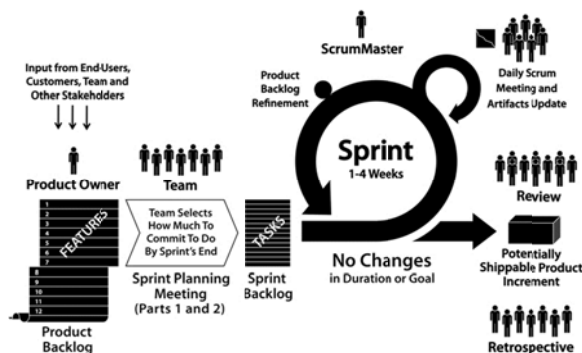


Figure 1. Key roles, artifacts, and principal events in SCRUM [4]

Sprint

Sprint phase is the one in which software development takes place. It consists of a selected list of requirements to be implemented in the next iteration, selected by the team together with the Scrum Master and Product Owner in the Sprint planning meeting.

Roles

In Scrum, there are three roles: The Product Owner, The Team, and The Scrum Master [4]:

The Product Owner: is responsible for maximizing return on investment (ROI) by identifying product features, translating these into a prioritized list, deciding which should be at the top of the list for the next Sprint, and continually re-prioritizing and refining the list. In some cases, the Product Owner and the customer are the same person.

The team: is responsible for building the product the customer will use, it is “self-organized” (self-managing), it has a high degree of autonomy and responsibility, and there is not a project leader. The Team decides what to commit to, and how to do the best to accomplish that commitment. It usually consists of seven people.

El SCRUM Master: is responsible for the SCRUM functioning in the project. He takes care of the aspects the organization needs, according to his knowledge and / or experience with the model. And also he takes care of those aspects that other people lack knowledge of. He makes sure everyone (including the Product Owner and those in management) understands and follows the practices of Scrum, and they help lead the organization through the often difficult change required to achieve success with agile development. The ScrumMaster protects the team from outside interference, and educates and guides the Product Owner and the Team.

SCRUM Meetings

SCRUM conducts the monitoring and the Project management through three meetings that make up the model [6]:

Sprint Planning: The bases for the planning are the client business needs and priorities. At this stage it is also determined which functionalities will the product incorporate after the next Sprint and how.

Sprint Monitoring: a brief, daily meeting, no more than 15 minutes, in which each member of the team explains the tasks they are working in, if any difficulty has aroused or if any is already anticipated, and updates upon the Sprint backlog the finished tasks, or the remaining work time.

Sprint Assessment: At the Sprint review meeting, the Team presents what was developed during the Sprint to the Product Owner and any other stakeholders who want to attend. After the Sprint review and prior to the next Sprint planning meeting, the ScrumMaster also holds a Sprint retrospective meeting in order to encourage the Team to revise, within the Scrum process framework, its development process to make it more effective and enjoyable for the next Sprint.

SCRUM Files

Within the generated files [9] and [4] are mentioned:

Product Backlog. It is a high level file for the Project, and it is evolving constantly. It contains generic descriptions of all the requirements, desired functionalities, etc., ranked according to its business value. It contains estimates, for the business value as well as the development effort required. **User Story** is a term very much used to designate the Product Backlog requirements or functionalities.

Sprint Backlog. It describes in detail how the team will implement the requirements during the next Sprint. Tasks are divided in hours, with a maximum of 16 hours. If this happens, tasks must be divided in a more detailed way. The tasks in the Sprint Backlog are not assigned but rather performed by team members the way they best consider it.

Burndown. It is a public graphic that measures the amount of unresolved requirements in the Project Backlog, at the beginning of each Sprint. The progress of the project can be seen when drawing a line that connects the dots of all finished Sprints. The common thing is that this line be descending (in cases in which everything goes right, in the sense that the requirements are well defined since the beginning and they do not vary) until it reaches the horizontal axis, in which case the project has ended (there are no more unresolved requirements in the Backlog). If during the process new requirements are added the straight line will have ascending slope in some segments, and if some requirements are modified, the slope will vary or even worth zero in some stretches.

SCRUM Phases

The methodology proposes the following three phases [7]:

Planning Phase:

Planning: the team, tools, and development system are defined. The Product Backlog is created as well as the list of known requirements together with their priorities, and the needed effort to perform it is estimated (Sprint Backlog).

Architectural Design: the architecture of the product is defined. It has to be one that permits to implement the requirements.

Development Phase: it is the agile part, in which the system is developed in Sprints. Each Sprint includes all traditional stages in software development: requirements survey, analysis, design, implementation and delivery.

Completion Phase: it includes integration, testing y documentation. It indicates the implementation of all the requirements, leaving the Product Backlog empty.

2. EXPERIMENT

Company origin

The methodology exposed above, was applied to a company, physically located in Corrientes province, offering software development services in the NEA region of the country. It was set up by the integration of professionals and advanced students from the careers

Graduates in Systems (UNNE¹) and Engineering in Information Systems (UTN²- FRR) in order to face together the first and unique job opportunity, to develop a full software system, that includes sales and client management, stock list, accounting and logistics, among others, for a furniture company from Formosa province. The company was originally composed of three founding partners, whom estimated that the project requested by the contractor, will consume a first developing stage of between 12 and 15 months, followed by a trial period of 6 months and the corresponding system maintenance.

Company's structural situation

Nowadays, the company is at the beginning of the creative entrepreneur stage, which first crisis was the emergence of a great deal of transactions that escape control without a formal tool.

The PyMES³ current conformation is interdisciplinary, formed by 2 (two) University Application Programmers, 2 (two) Graduates in Management and 3 (three) Engineers in Information Systems, whom through a rudimentary procedure intended to control their transactions. However, when business opportunities arise, and in order to make profit out of them, it is necessary to have the support of an organizational structure according to the requested characteristics.

Origin of the problem

After an organizational diagnostic, carried out by Management professionals, the following problems and their associated risks could be identified [2]:

Inadequate Organizational Structure: the risk arises due to the lack of experience and / or lack of management knowledge of the software enterprises founders. As a consequence these companies tend to remain stuck in a certain development and growth level in which the structure cannot bear the increasing trading volume.

Inefficient Project Management: This is linked with the software production area, where development is tailor-made. In this process, the production costs that are incurred have a high impact in total costs, mainly, developing hours (hours - man), which if not managed appropriately may provoke an important loss for the company. These represent 90% of total costs in most projects. Linked closely to this problem, it is the budgeting process, at this point it is vital to have an estimation as close as possible to the reality. Besides, there is an essential factor in project management: the team work; to find the optimum way to deal with it, flexibly and according to the installed capacity, which will be the key to success.

SCRUM selection reasons

Because of the stated above, the objective is: to strengthen the company's operative management, by means of using a formal tool, one that is adequate to the business profile and also that provides a more agile and flexible view with management principles and guidelines to innovation in its application.

Next, the main advantages that support the choice for this methodology over other existing ones are mentioned:

It is simple: it is easy to transfer knowledge to others, who will understand it thus being able to put it into practice. There is a need for a SCRUM Master who understands his role and who's willing to carry it out with dedication, and a team that will self-manage.

Emphasizes visibility: it is a vital factor of SCRUM that the progress achieved is visible, that is to say, that the deliverables be functional and probable. There is a key moment in which that characteristic reaches its peak, during the Sprint Review, moment in which the team presents the progress made in the last Sprint. On top of this, the client may decide to put into production the already developed stuff, and thus determine the beginning of the recovery of his investment.

It has clear rules and roles: Every activity and workflow is defined explicitly by the use of clear norms. The SCRUM Master role makes it possible and controls that these rules are obeyed.

It is based upon personal commitment: the team is committed to implement certain functionality in a certain time (generally 15 days duration Sprints). It does not have power over 'what' only over 'how much' (how much can be done in a Sprint) and 'how' (it is responsible for the technical solution).

It solves problems day by day: the daily SCRUM meetings are, in an effective way, good for detecting problems, carry out a daily monitoring, and planning to the short term, thus they permit to materialize and optimize the use of time.

It allows carrying out agile measurements: these measurements facilitate feedback. For example: team's developing speed can be measured; it is possible to compare progress time among different Sprints and projects that later facilitate and incorporate accuracy to the process of time estimation and budgeting.

SCRUM Implementation

After the organizational diagnostic carried out by specialists in Management, the process of SCRUM implementation started. It consisted in three stages:

1. In the first place, the development team suggested this methodology as the most adequate one to strengthen the company's projects management. This proposal was verified with the management report of the Management specialists. A training meeting in SCRUM methodology was arranged. In this meeting, the most important concepts and principles of the Agile Movement and a theoretical basis of SCRUM were exposed, and then the application concepts were both presented and discussed. After this stage, all members of the company had a clear understanding of the adopted methodology.
2. SCRUM Application to an ongoing Project: at this stage, the management team had to design and take over the role of SCRUM facilitators. They had the objective of guiding developers in the application; that is to say, apply all the conceptual contents learned, in an ongoing project in the company, thus accomplishing the theory-practice relationship in reality.
3. The innovation at this stage was linked to the fact that the role of SCRUM Master was not formally assumed by any developer, the idea was to encourage the team self-management and that every member should have knowledge of the responsibility that it implies. Thus it was decided that the mentioned role

¹ UNNE: Universidad Nacional del Nordeste – Facultad de Ciencias Exactas y Naturales y Agrimensura

² UTN: Universidad Tecnológica Nacional – Facultad Regional Resistencia

³ PyMES: Small and Medium Enterprises

was rotational among the members from one Sprint to the other.

4. Monitoring of SCRUM application in the company: it was supported by the management team in their facilitators role. A web service of shared storing was implemented in order to carry out the tracking of the project's documents (working hour charts, Burndown, etc.).

The Project

The study case in which the exposed methodology was applied, tackled the execution of an integrated management software system for a furniture factory network located in Formosa province.

The following roles were assigned to the participants:

- The Product Owner: one the case peculiarities is that the Product Owner is outside Corrientes province. This determined that the communication between him and the team was made through e-mail, Messenger, Skype, etc., supported by post-planning documentation and online reviews.
- The Project focused more towards self-management than to the SCRUM Master prevalence.
- Facilitator team (administrators): they were in charge of promoting the SCRUM application and of including a holistic view upon software development management.
- The development team members, constituted by 5 (five) Analysts-programmers: had an active role during the whole SCRUM application process.

Initial Backlog of the product

The Product Backlog was developed from interviews with the Product Owner and key people interested in the project; the main objective was to have all the necessary information about the Business Case. Then, it was created a list with all the requirements or functionalities ranked; these were obtained from an immediate need analysis of the business. This stage was characterized for the difficulty to conciliate objectives; manage different codes, from part of the company technicians and the Product Owner. Before the planning meeting of the first Sprint, the working team created a requirement list based on the needs exposed by the Product Owner.

Team's initial working speed

In order to estimate the team's working speed, first it was determined how many available hours each team member had during the 15 days of the Sprint [5]. A total of 101, 5 weekly hours was obtained, considering that the 100% of total hours are not worked entirely. It was decided to implement a dedication factor (percentage of effective work to be done). Its value was determined in 40% based on work experiences on current publications [5], that is to say, it was established an effective work time of 81, 2 hours for the 15 days duration of the Sprint. Thus the team agreed to carry out their tasks in a more appropriate way according to their real capacity.

Sprint Planning

It is the most important meeting in the SCRUM application; it is vital to devote the necessary time to it. Upon the reached conventions, deadlines, estimations and deliverables were defined. Its importance lies in the fact that it is the software development process starting point. The results are exposed next:

Sprint Duration

It was agreed to use Sprints of 15 working days during the whole project, such a time was defined as suitable because it had been decided to generate constant contact with the client and thus respond appropriately to the requested changes and to the possible problems that could arise.

Functionalities breakdown of the Product Backlog

The Product Backlog functionalities which were previously prioritized in agreement with the Product Owner were broken down based on the team's technical experience, in **concrete and manageable tasks**. The objective was to determine simple tasks that allow a practical assignment more convenient and that at the end of the Sprint there was a functional deliverable, visible to the client.

Tasks Estimation: Use of Poker Planning

After, breaking down functionalities into tasks, there was a general view of their volume. It was specified:

- Task Name.
- Work Order.
- Time Estimation.
- Technical comments for its execution.

In order to carry out the tasks estimation time, the Poker Planning technique was used: each member of the team had a pack of 13 cards (Figure 2). Each time that a task should be estimated, it was selected a card that represented its time estimation (in developing hours) and it was placed with the numbers facing down on the table. Once all team members selected their cards, these were turned up at the same time. Thus each member was forced to think for him/herself instead of following the others estimation. After a debate about technical opinions, the specific times for each task were agreed on. The estimations were register on an excel chart similar to that on Figure 3. The objective of this planning was:

- Involve each member in every task.
- Obtain each member own view without being affected by that of the expert.
- Solve conflicts prior to development. Exploring possible inconvenients and anticipating solutions.
- Discuss technical aspects in each task.
- Share experiences.

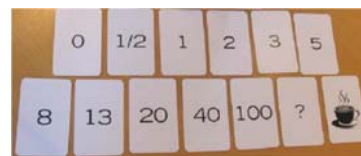


Figure 2. Planning Poker Cards

	Expected time
Product charge stock	26
Load BD	4
Design of the database	5
Bring order appropriate consultation	1
Print label	4
Programming	12
Labels	30
Technology Research	15
Design	4
Check database	1
Programming	7
Generate bar code	4
Print labels	4
Burden of purchase invoices	20
Orders get loaded	2
BD review	1
Design	5
Programming	12
Unplanned	
High orders screen	8
Unify cooperators	5
Modify queries	12

Figure 3. Return of scheduled tasks

Sprints Construction

The prioritized functionalities and their corresponding tasks (estimated in developing hours) were assigned to the first Sprint. To do this, the team’s developing speed and the Sprint duration were considered.

Creation of the Sprint board

With the all necessary information, the Sprint board was built; it is composed by 4 sections. The first section is about unfinished tasks, the second one, about the assigned tasks, the third section shows the finished tasks and the forth, represents the unplanned tasks necessary for the Sprint completion. In this board all the functionalities, their respective tasks and the Burndown graphic are shown.

Post its of different shapes, sizes, and colours were used for the comments and lists. These allow a dinamyc and agile use.

Daily Work

At the beginning of the day, the developers selected the tasks to be done, choosing freely, according to their importance from the unfinished task group. Each one chose the task they felt more comfortable with, writing their names and putting them onto the completed tasks as they finished. In each daily meeting the information on the board was updated.

Sprint Completion

At the moment of ending each Sprint, there was a module of the final deliverable product. Besides, there was a meeting of retrospective. In that meeting the team discussed the Sprint results, analyzing the Burndown graphic (Figure 4), determining the changes to be done onto the next one, as to increase the productivity, correcting inaccuracies and suggesting improvements.

The biggest detected problem was linked to the underestimation, in working hours, of the broke down tasks in the first Sprints, testing, changes in the data base, correction, and functionalities demonstration. This was due to the team’s lack of experience in estimation. This led to delays in tasks completion.

On top of this, there was inconvenient when solving problems derived from:

- Lack of information about the business logic when facing a problem.
- Lack of communication within the team.

This problem was tackled down by the implementation of a virtual information storage unit, which was as a common space for exchange and update. More precisely, it was Dropbox, in its free version, the tool that was used. Dropbox is a file hosting service that offers cloud storage, online file synchronization as well as among computers and it allows to share files and folders with others. Once the desktop client programme was installed, it was possible to leave any file in a given folder. That file is synchronized in the cloud and in all the other desktop units of the Dropbox client.

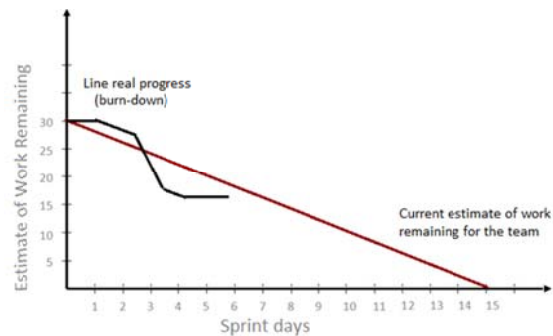


Figure 4. Sprint Burndown Chart (Partial Sprint)

3. CONCLUSION

The interdisciplinary links among Graduates and Engineers in Systems and Graduates in Administration allowed to face SCRUM application not only as an isolated practice within PyMES and for software development, but also as an action inside the strategic plan of it. The main objective was to initiate a formalization process in projects management and to develop a dynamic organizational structure that will dispose the company for future growing phases.

It is considered that the experience of using SCRUM was ultimately positive. A complete and adapted application of all the practices was carried out, involving the organization actors comprehensively. The planning mistakes made during the first Sprint served as an important feedback in order to replan the followings. The first stages meant an increase in the productive efficiency, proving that it is possible to apply a methodology that would speed up management and optimize costs, and also generate software with dynamic requirements.

One of the most evident outcomes of the application of this practice was the involvement of the people with the project and with the company’s objectives; exteriorized through a high level of commitment and participation from part of each member in the planning, design and execution stages.

The fact of rotating the SCRUM Master role may be favorable for the future as it encourages independence in case of absence of any participant whether temporarily or permanently, allowing reassignment of the role.

On the other hand, the Burndown graphic allowed observing the improvement in the estimation, contributing to facilitate the completion of any action when reality deviates from planning. Besides, it gives the team an idea about its status as regards commitment.

After the continuous application of the exposed model, it could be appreciated that when the team moves forward faster than expected, it must consider, in the Sprint planning stage, the estimation of more tasks in order to continue working. Likewise, if the team falls behind, because of a wrong estimation of the tasks, there is the possibility to cancel the Sprint and to plan again, this time with more accurate data.

Experience was acquired in the breakdown or itemization of tasks, and also the team's communication capacity was powered.

The process of adapting this practice to the company's own structure and culture was a right decision that will allow in the midterm, adopting SCRUM fully, including it to the organization in a natural way.

4. REFERENCES

- [1] Canós J., Letelier P. y Penadés M. (2010): "Metodologías Ágiles en el Desarrollo de Software". Universidad Politécnica de Valencia. En: <http://www.willydev.net/descargas/prev/TodoAgil.pdf>.
- [2] Chiavenato H. (2006). Introducción a la Teoría General de la Administración.
- [3] Coplien, J. (1994). "Borland Software Craftsmanship: A New Look at Process, Quality and Productivity". Proceedings of the 5th Annual Borland International Conference, Orlando, Florida
- [4] Deemer P., Benefield G., Larman G. y Vodde B. (2010): The Scrum Primer. Scrum Training Institute. In <http://assets.scrumtraininginstitute.com/downloads/1/scrumprimer121.pdf>.
- [5] Henrik K. (2007). Traducción Proyectapolis: Scrum y XP desde las Trincheras, Cómo hacemos Scrum,
- [6] Mahnic V., Vrana I. (2007). Using stakeholder-driven process performance measurement for monitoring the performance of a Scrum-based software development process.
- [7] Mendes Calo, K., Estevez, E. y Fillotrani, P. (2011): "Un Framework para Evaluación de Metodologías". En: <http://www.egov.iist.unu.edu/cegov/content/download/1878/47500/version/8/file/Un+Framework+para+Evaluacion+de+Metodologias+Agiles.pdf>.
- [8] Oliveros, A. (2007). Curso Administración de Proyectos de Software. Maestría en Ingeniería del Software. Universidad de La Plata
- [9] Palacio J. (2008). Flexibilidad con Scrum.
- [10] Palacio, J. and Ruata, C. (2010) "Scrum Manager Proyectos". In http://www.scrummanager.net/files/sm_proyecto.pdf.
- [11] Sutherland J. (1996): ScrumWeb Home Page, A Guide to the SCRUM Development Process Jeff Sutherland's Object Technology Web Page.
- [12] Schwaber K. (1996): "Controlled Chaos: Living on the Edge." American Programmer, April 1996.
- [13] Sommerville, I. (2006) Ingeniería del Software, Séptima Edición.
- [14] Sutherland J., Schwaber K. (2011). The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework, in <http://jeffsutherland.com/ScrumPapers.pdf>
- [15] Takeuchi H. y Nonaka I. (1986). "The New Product Development Game." Harvard Business Review.