A Practical Approach to Project Management in a Very Small Company

Edgar Caballero and Jose A. Calvo-Manzano

Departamento Lenguajes y Sistemas Informáticos e Ingeniería del Software
Universidad Politécnica de Madrid,
28660 Boadilla del Monte, Madrid, España
ecaballero@bolesfactory.com,
josentonio.calvomanzano@upm.es

Abstract. This article shows how a very small company has tailored Scrum according to its own needs. The main additions made were the "sprint design" phase and the "sprint test" phase. Before the sprint 0, the requirements elicitation and the functional specification were made in order to meet deadlines and costs agreed with clients. Besides, the introduction of an agile project management tool has supported all the process and it is considered the main success factor for the institutionalization of the Scrum process.

1 Introduction

Due to the global economic crisis, organizations are forced to adapt their business strategies in order to stay in the market [1]. Spain and particularly the software industry are not the exception. Moreover, an inappropriate quality and project management in software organizations generate cost overruns, low quality and cancelled projects [2]. Organizations must improve their competitiveness through improvements in productivity in order to survive in a market weakened by the crisis [1].

At this scenario, organizations need a more efficient resources management. The goal is producing more in less time with the same or less cost is a key factor. A process improvement is necessary to achieve this goal [3], but a very small organization does not have enough time or resources to invest in it [4].

Several process improvements (e.g. "COMPETISOFT" [5]) have been proposed for small organizations based on defined process control [6] and empirical process control [7]. Agile methods are based on empirical process control and provide a good performance in small teams [7]; therefore they are suitable for process improvements in very small organizations.

This article shows how a very small company has implemented an agile method such as Scrum and how it was adapted, supported by an agile management tool, according to its own needs.

In the following sections it will be described the context, the tailored process and the agile management tool. Finally, conclusions will be summarized.

2 The Context

2.1 The Organization

Bolesfactory is a very small company of software development with a staff of 14 people [8].

At 2010, due to the actual crisis, the organization started to get worried about productivity and quality. In order to improve them, the organization got interested in agile methodologies and decided to introduce one of them. After evaluating several alternatives (Scrum, Extreme Programming, Kanban), the management decided to use Scrum.

For the past 3 years the organization has been working with Scrum and has some lessons learned about it, mainly with the problems detected.

Initially Scrum was applied without any change, but after some experiences, the organization began to make adjustments according to the needs that were emerging.

As in previous experiences with other methodologies (e.g., Team Software Process), productivity was affected by the lack of specialized support tools in the implementation of the process. For this reason, the organization decided to use an agile project management tool that achieves the expected levels of productivity by improving the visibility of the project.

2.2 Scrum

Scrum is an agile process framework that allows organizations be focused on business value through the frequent and regular delivery of high quality software [8].

Scrum is based on an empirical process control model rather than the traditional defined process control model, which regularly inspects activities to monitor what is happening and adapts them to produce the desired and predictable outcomes [9].

Scrum allows the team to apply any specific method or technique oriented to the software development.

The main criteria for selecting Scrum were:

- Several characteristics and principles were similar to the previous iterative process of the organization (Team Software Process).
- Hypothesis: Scrum increases productivity and reduces time to benefits [10].
- Scrum leads as the most adopted agile methodology [11].

Scrum implements an iterative and incremental process which involves three stakeholders: the Product Owner, the Team, and the ScrumMaster.

The Scrum process defines an initial preparation phase and several iterations called sprints (see Fig. 1). A sprint is a 2-4 weeks period of development time and 4 meetings are held: planning, daily, review and retrospective meetings.

The requirements are collected and prioritized in the product backlog, which is decomposed in tasks on the sprint backlog.

The planning meeting is decomposed in 2 sessions. The first session is focused on explaining the sprint scope, and the second session is focused on tasks identification and effort estimation.

During the execution of each sprint, the team meets daily in the 15-minute meeting to track the work progress answering three questions: What have I done since the last Scrum meeting? What will I do before the next Scrum meeting? What prevents me from performing my work as efficiently as possible? [10].

Fig. 1 shows the Scrum process.

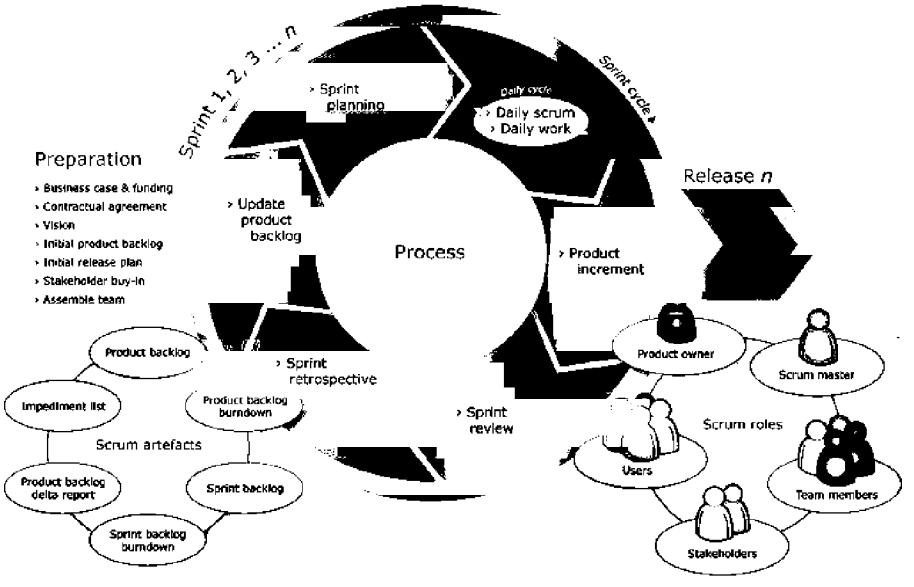

Fig. 1. Scrum process

2.3 Agile Management Tool

The tool selected by the organization was "Target Process" [12]. Its usability and flexibility were the main criteria for the choice.

Target Process is an agile project management software designed with simplicity in mind, Target Process helps software development companies to reduce the complexity of software project management and simplifies planning, tracking and quality assurance activities.

Target Process is a customizable tool. You can create a development process with customizable practices, workflows, terminology and customized fields.

3 Tailoring Scrum in the Organization

After introducing Scrum in some projects, the team members were motivated with the Scrum project management strategy. However the organization detected several problems such as:

- A high number of changes because a low initial requirements definition.
- A low quality in the product during the "sprint review".
- An unstable product after each sprint caused by the incremental strategy.
- Many wasted hours in getting the project graphics and reports.

In order to solve the previous problems, the organization began to define additions that will be described in the next sections.

3.1 The Tailored Process

In this section, the sprint flow will be described in order to understand the tailored process. Fig. 2 shows the sprint flow highlighting the additions.

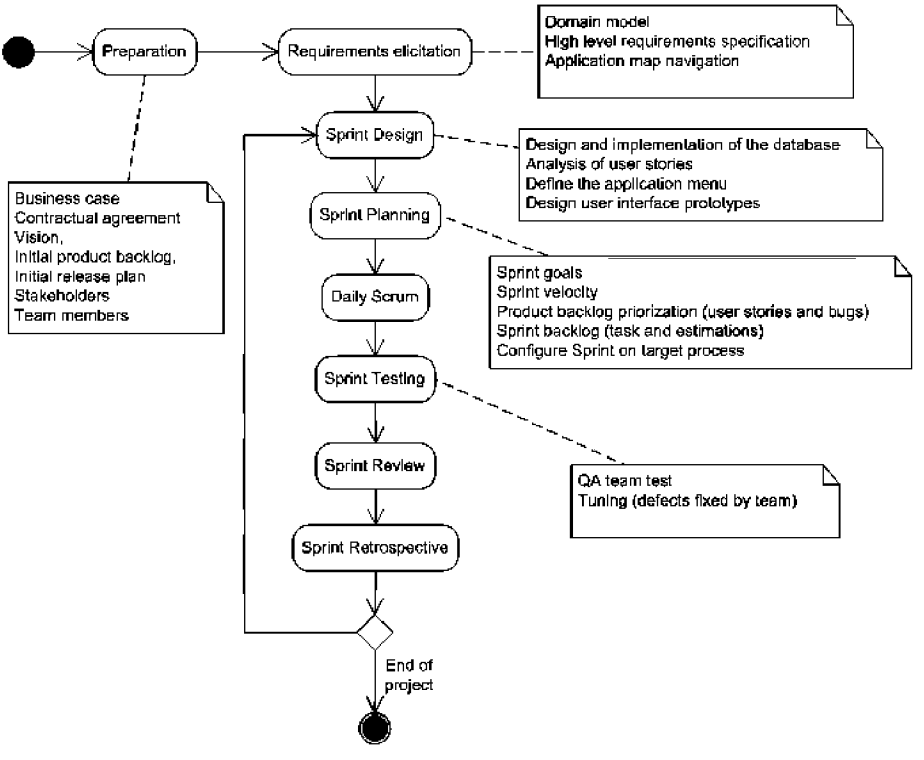

Fig. 2. Sprint flow chart

As shown in Fig. 2, “requirements elicitation”, “sprint design” and “sprint testing” are the additions introduced in the Scrum process.

After the “preparation phase”, the process continues with the “requirements elicitation”. Its goal is to get the domain model, the high level requirements specification and the application map navigation in order to improve the requirements understanding and their cohesion.

The iterative process starts with the “sprint design”. It is focused on increasing the knowledge for improving the estimations during the “sprint planning”, and it includes the data base implementation, the analysis of user stories, the menu application and the main user interface prototypes.

Then the team continues with the “sprint planning”, in which only the second session takes place (tasks identification and effort estimation). The first session with the “product owner” is moved to the “sprint design”.

After that, the “daily scrum” begins. It includes the coding phase and the tests performed by the development team.

Before the “sprint review”, the “sprint test” is started. At this point, the quality team performs the integration test and the product verification. If there is time, the development team uses this time to fix major defects found by the quality team.

Finally, the “sprint review” and the “sprint retrospective” are done without any modification as described by Scrum.

Fig. 3 shows the tailored Scrum process.

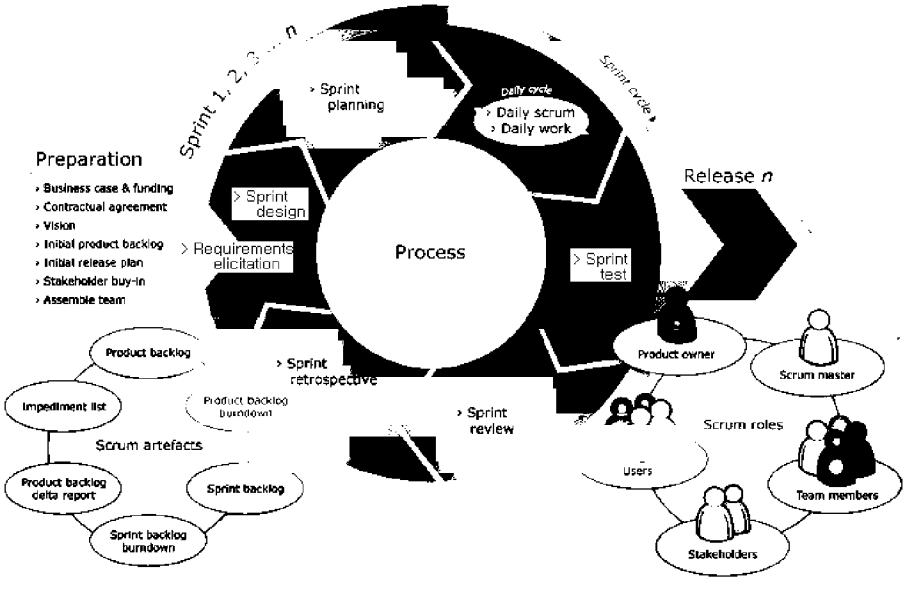

Fig. 3. Tailored Scrum process

3.2 Addition 1: The Requirements Elicitation in the Preparation Phase

The preparation phase (also called sprint 0 or pre game phase) organizes and defines all project needs for starting the Scrum iterations (sprints), but Scrum does not say how to do it. Its goal is to stablish the project scope delimitation and it contains the business case, the contractual agreement, the vision, the initial product backlog, the initial release plan, the stakeholders and the team members.

According to the organization experience, the preparation phase and the first planning session were not enough to get a good understanding of the project, the requirements individually were stable, but low cohesion between them forced to make frequent changes to the initial definition of requirements.

Scrum is only a project management framework and it does not specifies a requirements management method. Each organization has to select one.

At this scenario, the organization decided to introduce a requirements elicitation method in order to get a high level functional specification which includes the domain model, the high level requirements specification and the application map navigation.

This high level functional specification has two goals:

- Improve the requirements cohesion.
- Reduce the deviation of the project in order to meet deadlines and costs agreed with clients.

Based on this functional specification, the prioritized product backlog is created.

3.3 Addition 2: Sprint Design

Scrum suggests that the tasks of analysis, design, implementation and testing are performed during the execution of the sprint, but the decision on how to implement them is up to the development team. In this sense, the organization decided that the analysis and design were implemented before the "sprint planning" meeting in order to improve the requirements knowledge and reduce the estimate deviations.

"Sprint design" is approached as a systematic activity to improve the knowledge of the project before starting to code. Like others scrum activities, the "sprint design" must be a time box activity, in the case of a 4 week sprint, the "sprint design" should last less than 3 days.

The "sprint design" has the following tasks:

- Design and implementation of the database.
- Analysis of user stories.
- Define the application menu.
- Design the user interface prototypes.

Therefore the "sprint design" is not a scrum modification, but a specific addition detailing an aspect not described by the methodology that may or may not be used by another organization.

The user interface prototypes are an excellent tool for an early user validation [12]. Fig. 4 shows an example of a user interface prototype (left side in Fig. 4) designed with Balsamiq Mockups [13], which is very similar to the final implementation (right side in Fig. 4). Their main advantage is the short time needed for its design.

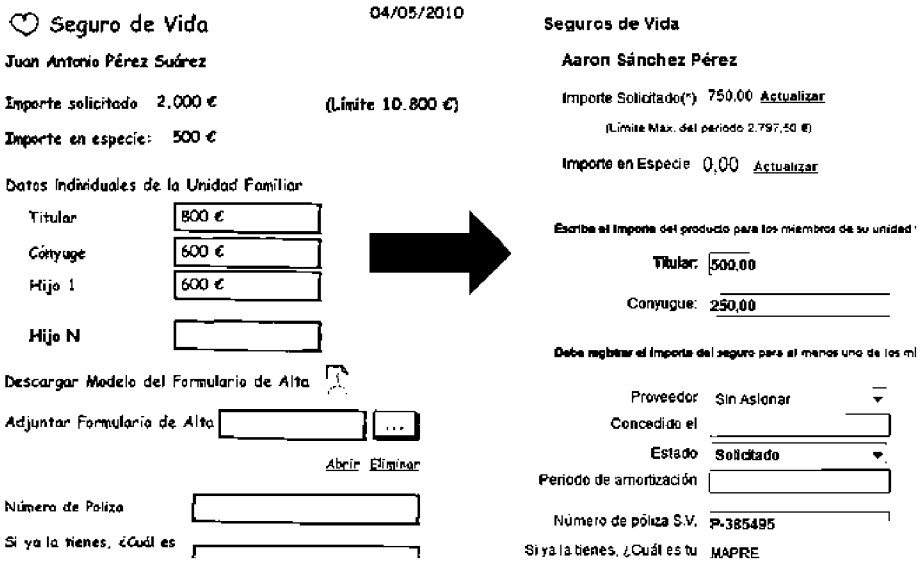

Fig. 4. User interface prototype

3.4 Addition 3: Sprint Test

The product quality after each sprint was not satisfactory. According to the good practices described in [15], the organization decided to create a quality team in order to test the user histories. After some sprints, it was observed in the "sprint review" that the user stories implemented at first had a better quality than the others implemented at the end, especially those that need an integration job.

The conclusion reached was that the user stories with bad quality were not tested enough, mainly because the sprint end date was reached, and often the team changed to "done" the user stories state when in fact they were not. Initially it was decided to allocate more time for the integration testing during the sprint, but it turned curiously detrimental from the standpoint of the project monitoring because the estimated effort for the integration testing distorted the project status during the early days of the sprint. This small and insignificant detail became a factor of discouragement for the team because the sprints ended with delays that were not identified until late in the sprint. To solve this problem, integration tests were separated from development by creating a new phase called "sprint test" to be performed before the "sprint review". The effort allocated for this activity is 20% of the effort that was initially assigned to each sprint.

The "sprint test" is performed by the quality team. During this phase, the development team can also fix the higher priority defects in order to improve the product quality at the "sprint review". This fixing activity is called "sprint tuning" and must be adjusted to the time box established for the "sprint test". The "sprint test" improves product quality but reduces productivity because it is a failure quality activity [16].

4 The Support Tool for Agile Project Management

The tool selected by the organization was "Target Process" [12]. Its usability and flexibility were the main criteria for selecting it. It is free for small teams (upto 5 members). In the next sections the activities supported by the tool will be described.

4.1 Change Management

Although in agile methodologies, the change management process is open and does not require formal approval, it is important to have at least a record of change requests and whoever has done. With "Target Process" the user is the one that registers the request through an email, the tool automatically classifies the request and finally the "product owner" updates its priority.

4.2 Bugs Management

"Target Process" allows tracking all the defects found by the quality team or users. The users can add, prioritize, plan iterations and view quality reports. In the bugs list, users can change bug states, assign bugs to developers and testers and edit bug (change severity, release/iteration, effort, etc.).

4.3 User Histories Management

"Target Process" allows storing requirements as features or as user stories. The user stories can be created on the fly or be imported from a .csv file. A user story could be assigned to two team members in the case of pair programming. One interesting feature is the storage of the initial estimation in order to analyze effort deviations.

4.4 Impediments Management

An impediment (or block) is an obstacle that prevents the sprint completion. "Target Process" provides the functionality needed to manage impediments. It allows adding impediments to user stories, tasks and bugs in order to make them visible for all team members.

4.5 Sprint Planning and Load Balancing

The planning and load balancing is one of the main characteristics in "Target Process". For planning, the user must establish the team members, their velocity, the start day, the duration and the sprint velocity. After that, the tasks are identified for every user story, the team members are allocated and the work load is estimated. Then, from the load balancing panel the user verifies if the user stories and bugs selected for the current sprint are according to the sprint velocity. If the estimated

effort that is required for the sprint does not exceed the sprint velocity, the work load of the team members must be verified. If some of them are overloaded, then their tasks or bugs must be reassigned.

4.6 Project Monitoring

"Target Process" supports the project monitoring through the burn down chart, which is drawn automatically considering the initial workload, the variation from baseline (change control) and the projection of the work to do according to the average velocity attained until the day before. The burn down chart shows the sprint progress detailing the iteration progress, the ideal line and the forecasted progress. It only requires that every day the team members update the time remaining for each task in which they had worked during the day.

Fig. 5 shows an example of a sprint burn down chart collected from one of the projects developed by the organization.

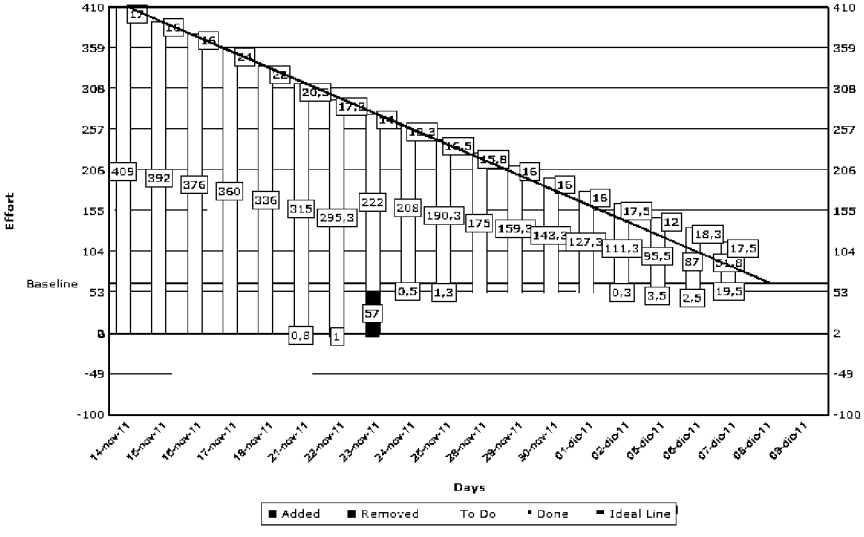

Fig. 5. Sprint burn down chart

4.7 Daily Activities Support

The tool provides two utilities that facilitate teamwork, even if the team members are located in different places.

1. "Team board" allows performing the daily meetings. It shows what is in progress or what was finished yesterday by any team member.
2. "Task board" allows following the bugs and task state. The team members can drag and drop the items in order to update their states.

5 Conclusions

Scrum is a methodology that seems simple but requires discipline and above all ability and experience of the development team. Similarly, as described by its authors, Scrum meets the needs of project management but does not address other important processes such as requirements management.

Therefore, to implement Scrum an organization must make additions based on their own needs and experiences to enhance the process, at least during the first months.

The following are the main conclusions from the study:

- Organizations must be careful not to alter the basic principles of the methodology.
- The definition of requirements achieved with Scrum was not enough to meet deadlines and costs agreed with the clients.
- It is necessary to establish some discipline with some activities because the team was not able to be self-managed or self-organized as Scrum requires.
- An expert is necessary when the team is new in the process.
- To achieve the productivity indicated by Scrum, the team needs a previous experience and excellent technical skills.
- A project management tool with Scrum features must be incorporated.
- The organization must be patient in order to reach a high productivity level and avoid modifying the heart of Scrum. There is a high probability of cancellation in the first sprints.

Some benefits identified with the additions introduced to Scrum are:

- Adjust the sprint tasks for improving the estimation.
- Reduce the ambiguity of the project requirements.
- Allow users to validate the user interface prototypes.
- Minimize the probability of changes.
- Increase the product quality with a specific test phase.

5.1 The Unfinished

The instability of the product due to increments generated by the sprints is the highest priority in the process improvement inside the organization. Different techniques have been identified to improve this aspect but they have not yet been implemented. Adopting an agile methodology requires to be prepared not only technically but also mentally, because it is passed to a model where the change in a user requirement is not a trouble. This is logical when the value provided to the customer is what prevails over any other criteria, and in fact is a competitive factor. But not all organizations are ready to assume the economic consequences of these changes when the impact is not easily measured and cannot be controlled. Continuous changes and increments generated by the sprints require the establishment of a regression test strategy. Applying Scrum specific practices should be strengthened including some "extreme programming" techniques. In this sense, the organization is currently working on "test driven development" (based on unit tests) and "continuous integration" (in order to

support regression testing). These techniques will improve code quality and will reduce the probability of destabilizing the product due to frequent deliveries and continuous changes. Once these techniques will be implemented, the next step is the automation of the user interface testing and code generation to continue increasing productivity.

Acknowledgements. This work is sponsored by everis Foundation and Universidad Politécnica de Madrid through the Research Chair in Software Process Improvement for Spain and Latin American Region.

References

1. Schwab, K.: The Global Competitiveness Report 2010–2011. World economic forum (2010), http://www3.weforum.org
2. Standish group: CHAOS Manifesto (2011), http://standishgroup.com/newsroom/chaos_manifesto_2011.php
3. Glazer, H.: Time to Market vs. Process Discipline. In: SEPG Conference, Nashville, TN, USA (2006)
4. Richardson, I., Wangenheim, C.: Why are Small Software Organizations Different?, Guest editors' introduction. IEEE Software 24(1), 18–22 (2007)
5. Oktaba, H.: Software Process Improvement: The Competisoft Project. IEEE Computer Magazine 40(10), 21–28 (2007)
6. Pino, J., García, F., Piattini, M.: Software process improvement in small and medium software enterprises: a systematic review. Software Quality Journal 16(2), 237–261 (2007)
7. Tore, D., Torgeir, D.: Empirical studies of agile software development: A systematic review. Information and Software Technology 50(9-10), 833–859 (2008)
8. http://www.bolesfactory.com
9. Schwaber, K.: Agile Project Management with Scrum. Microsoft Press (2003)
10. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall (2002)
11. West, D., Grant, T.: Agile Development: Mainstream Adoption Has Changed Agility. Forrester Research (2010)
12. http://www.targetprocess.com
13. Kamalrudin, M., Grundy, J.: Generating essential user interface prototypes to validate requirements. In: 26th IEEE/ACM International Conference on Automated Software Engineering, USA (2011)
14. http://www.balsamiq.com
15. Kniberg, H.: Scrum and XP from the trenches. InfoQ (2007)
16. Humphrey, W.: A Discipline for Software Engineering. Addison-Wesley (1995)