# Introducing agile software development: lessons learned from the first scrum project in a Slovenian company

**Conference Paper** · January 2015

DOI: 10.2495/AMEIT140781

**4 authors**, including:

Tomaž Hovelja
University of Ljubljana
**15** PUBLICATIONS **88** CITATIONS

SEE PROFILE

Viljan Mahnic
University of Ljubljana
**63** PUBLICATIONS **258** CITATIONS

SEE PROFILE

# Introducing agile software development: lessons learned from the first Scrum project in a Slovenian company

N. Zabkar, T. Hovelja, J. Urevc & V. Mahnic
*Faculty of Computer and Information Science, University of Ljubljana, Slovenia.*

## Abstract

Agile software development using Scrum is an important research issue in the area of software engineering. In this paper, we describe one of the first Scrum projects used in Slovenian companies, which served as a case study for the analysis of some typical activities concerning Scrum introduction to an industrial environment. Three aspects of this project are presented: (1) the most important preparatory activities, (2) the project progress monitoring procedures, and (3) the accuracy of effort estimates. These aspects are analysed on the basis of empirical data gathered during project observation. The most important findings are summarized in the lessons learned in order to increase the knowledge about Scrum adoption and help other companies that plan to introduce Scrum to their development process.

*Keywords: agile software development, project management, Scrum, effort estimation, user story, planning poker.*

## 1 Introduction

Agile methods [14] have been gaining wide acceptance among mainstream software developers. The results of the Standish Group's 2011 Chaos Report show that the success rate for agile projects (42%) is three times higher than for waterfall projects (14%). According to the last State of Agile Development Survey [12] the most widespread agile method is Scrum [10, 11]. Scrum or Scrum variants are reported to be used by 73% of respondents. Nevertheless, the number of studies on Scrum (e.g., [5, 6]) still remains scarce [3].

The aim of this paper is to contribute to empirical evidence of Scrum adoption by providing the experience of a company, which introduced Scrum within the scope of the project of a complex media portal development. The remainder of the paper consists of the case study description (Section 2) and critical evaluation of three aspects of the project studied: initial preparatory activities (Section 3), progress monitoring (Section 4), and accuracy of effort estimates (Section 5). Section 6 describes the most important lessons learned and recommendations, while Section 7 provides a conclusion.

## 2 Case study

### 2.1 Project background

The aim of the project was to rebuild a complex media portal (one of the largest in Slovenia). The use of Scrum was proposed by an external member of the development team who got acquainted with Scrum during a capstone course he attended at university [7]. Scrum was considered an appropriate development method due to vague and changing requirements, the project's nature and short time to deliver. The faculty members were allowed to observe the development process during the first seven Sprints, after which the management expected the first release to be finished, and to publish the results for research purposes. This deadline was based primarily on the management's expectations rather than on a realistic release plan. The number of people working on the project varied slightly from Sprint to Sprint between six and eight. The Product Owner and the ScrumMaster were full time regular employees, while the development team was mixed and consisted mostly of outsourced freelance programmers.

### 2.2 Typical Sprint

The company set the Sprint length to three weeks. Each Sprint started with a Sprint planning meeting on Thursday and ended with a Sprint review and Sprint retrospective meetings on the Tuesday of the Sprint's third week.

The Sprint planning meeting consisted of two parts. During the first part of the meeting the user stories with the highest priority were allocated to the Sprint so that the sum of story points fitted within the capacity determined by the Team's velocity estimate. During the second part, the Team decomposed each story into more detailed tasks and assigned responsibilities for each task.

Daily Scrum Meetings were held regularly every day. Each team member reported what he had done, what he planned to do, and what impediments were in his way. Additionally, each team member had to record the amount of work spent and the amount of work remaining for each task he was responsible for.

The Sprint review and the Sprint retrospective meetings took place on the same day. At the Sprint review meeting, the finished stories were demonstrated to the Product Owner and other stakeholders. At the Sprint retrospective meeting, the Team decided which practices should be improved during the coming Sprint in order to help them work better together.

# 3 Preparatory activities

Special attention was devoted to the assignment of Scrum roles, the definition of the concept of “done”, the choice of the Sprint length, and the choice of appropriate methods for project planning and performance monitoring.

## 3.1 The Product Owner role

The company was strongly advised to look for a person capable of communicating the vision of what is to be developed, maintaining the Product Backlog, and defining the criteria by which the results will be judged. Additionally, it was made clear that the Product Owner should provide timely answers to questions on the details of user stories, and make quick evaluations of the work being done. For all these reasons, it was very important that the associate editor of the media portal contents accepted the role of Product Owner. He had good knowledge of the media portals functionality and held a position high enough in the management hierarchy structure to direct the project in accordance to the company’s vision.

## 3.2 The ScrumMaster role

The ScrumMaster should be knowledgeable about software development methods and have enough power to ensure smooth running of a project. Considering the high priority of the project, this role was assigned to the head of the web development department, who had strong background in information systems. He accepted this role; however, knowing that he would not be able to be with the Team every day due to other obligations, he appointed a deputy to perform the ScrumMaster role in his absence.

## 3.3 Concept of “done”

A precise definition of “done” was developed before the start of the project, which required that each user story passed all acceptance tests, the code was written in accordance with coding standards, each software component was adequately commented, appropriate documentation was written, the code was peer reviewed by another member of the Team, the functionality of the user story was approved by the ScrumMaster, and the user story was accepted by the Product Owner.

## 3.4 Sprint length and Scrum meetings

The Sprint length of 3 weeks was chosen as a compromise between flexibility and overhead. With regard to Scrum meetings, the company decided to strictly follow Scrum rules and perform all prescribed ceremonies. In accordance with these decisions, each Sprint was conducted as described in subsection 2.2.

### 3.5 Project planning and performance monitoring

The planning poker method [2, 8] was chosen for estimating the effort required for the implementation of each user story. It was decided to use predefined values of 0.5, 1, 2, 3, 5, 8, 13, and 20 story points. A story point was considered to represent a working day consisting of 6 hours of effective work. For the purpose of planning and performance monitoring, the concept of velocity was introduced. The velocity estimate for the first Sprint was simply computed as a product of the number of working days in the Sprint and the weighted number of developers. For subsequent Sprints, it was agreed that the velocity estimates would be based on the actual velocity achieved in previous Sprints. Unfortunately, the project started before details about release planning were defined and the role of burndown charts was fully understood.

## 4 Progress monitoring

Although the project was considered a great success from the business viewpoint, the post-mortem analysis identified progress monitoring as a high priority topic offering most improvement opportunities. Only the difference between the planned and actual velocity was strictly monitored, while release and Sprint burndown charts were not used. The corresponding charts in Figures 1 to 3 were reconstructed by the authors after the end of the project in order to illustrate the deficiencies discussed in this paper.

### 4.1 Velocity tracking

As shown in Table 1, the difference between the planned and the actual velocity was rather small in Sprints 1 and 2, but increased significantly in subsequent Sprints. The actual velocity matched the planned in Sprint 7.

In Sprint 5, two new developers were added to the Team, creating a disruption which decreased the productivity of other team members. In Sprint 6, a substantial difference between the planned and actual velocity was the consequence of an overly optimistic velocity estimate. Instead of adapting the estimate to the actual achievement in the previous Sprints, the Team succumbed to the pressure of the approaching deadline and promised to deliver more functionality than actually possible. In Sprint 7, the Team set a realistic estimate of expected velocity and managed to deliver all of the committed functionality.

Table 1: Planned and actual velocity.

| Sprint | Planned velocity [story points] | Actual velocity [story points] | Difference [story points] | % of story points accomplished |
|---|---|---|---|---|
| 1 | 32.5 | 30.5 | 2 | 93.8 |
| 2 | 34.5 | 32.5 | 2 | 94.2 |
| 3 | 34 | 24 | 10 | 70.6 |
| 4 | 40 | 31.5 | 8.5 | 78.8 |

| | | | | |
|---|---|---|---|---|
| 5 | 39.5 | 7.5 | 32 | 19.0 |
| 6 | 42 | 29 | 13 | 69.0 |
| 7 | 32 | 32 | 0 | 100.0 |

### 4.2 Release plan and completion date

The initial Product Backlog was incomplete and new stories were constantly being added by the Product Owner. Even though this situation is not uncommon for agile projects, the project management did not consider the use of a release plan, which would enable the determination of the number of Sprints and an approximate completion date. Fig. 1 clearly shows that the Team was not able to reduce the amount of work remaining as quickly as the Product Owner added new stories to the Product Backlog. After seven Sprints, it became evident that the desired content of the release should be reexamined in order to get an acceptable completion date. It would be beneficial for the future Scrum projects to have the release burndown chart created and maintained at the very beginning of the project, so that the gap between expected and actual scope of work would be transparent at a much earlier stage.

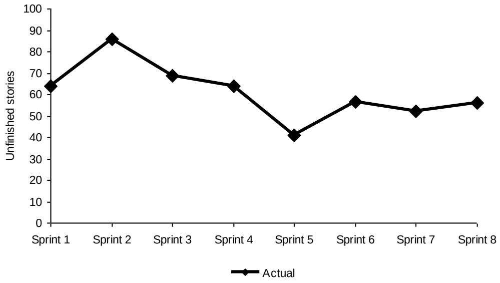

Figure 1: Release burndown chart at the beginning of Sprint 8

### 4.3 Sprint burndown charts

The findings regarding the release burndown chart can also be applied to the use of Sprint burndown charts. Figures 2 and 3 exhibit an extraordinary increase of the remaining work in the fifth (Fig. 2) and eleventh (Fig. 3) day of the Sprint.

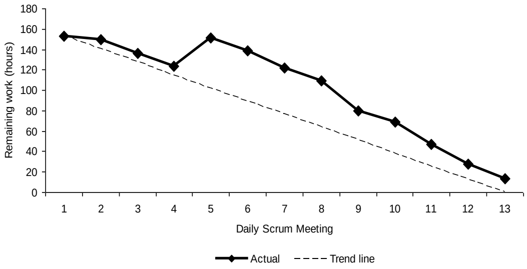

Figure 2: Sprint burndown chart for Sprint 1

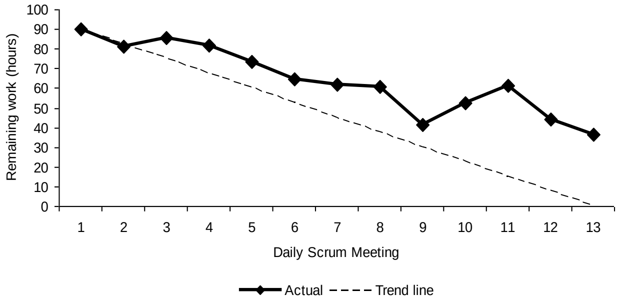

Figure 3: Sprint burndown chart for Sprint 3

This happened because some of the big user stories were not decomposed into tasks at the Sprint planning meeting (as required by Scrum), but later during the Sprint. Consequently, the Sprint burndown diagrams indicated a false sense of being not much behind the schedule instead of showing the real difference between the actual and the trend line.

## 5 Estimation accuracy

The aim of the detailed analysis of the estimation accuracy was to further investigate whether the planning poker estimates are more accurate than statistical combination of individual estimates as suggested by [9].

### 5.1 Study design

The analysis included all the stories added to the Product Backlog from Sprint 2 onward. The Team estimated these stories using planning poker and the estimates given by the Team members during the first round were recorded in order to compute the statistical combination. The statistical combination was computed simply as the arithmetic mean of the individual estimates, which according to [4] often works as the best method for combining estimates.
In order to calculate the estimation accuracy, planning poker estimates and statistical combination of individual estimates were compared to actual effort spent. The balanced measure of relative error (BRE) was employed, which is calculated as:

BRE = |actual effort – estimated effort| / min(actual effort, estimated effort)

The decision to use BRE as the key measure arose from the fact that BRE is a balanced measure, evenly balancing overestimation and underestimation. Because our BRE values exhibited deviations from normality in terms of skewness and kurtosis, the non-parametric Wilcoxon singed-rank test [13] was used to analyze the distributions of the two tested variables, and Cliff's delta [1] was applied to measure the size of effect.

### 5.2 Results

In total 49 user stories were estimated. In Table 2, the mean and median values of story size (in story points) and BRE are shown for each Sprint and each kind of estimates separately. The data indicate that (1) in all Sprints the mean BRE of planning poker estimates was lower than the mean BRE of statistical combination, (2) in all Sprints except one, the median BRE of planning poker estimates was also lower than the median BRE of statistical combination, and (3) the mean BRE of both kinds of estimates exhibited a falling trend, diminishing from Sprint to Sprint.

In order to test the hypothesis that planning poker estimates are more accurate than statistical combination, we compared the BRE for all of the 49 stories. The results of the Wilcoxon signed-rank test are shown in Table 3. The *p*-value of 0.059 indicates an almost significant difference in favor of planning poker; however, it is slightly too large to confirm the aforementioned hypothesis. The effect size of the difference can be considered small (Cliff's delta 0.171).

Table 2: Mean and median values of story size (in story points) and BRE.

| Sprint | Number of stories | Planning poker | | | | Statistical combination | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Story size | | BRE | | Story size | | BRE | |
| | | Mean | Median | Mean | Median | Mean | Median | Mean | Median |
| 3 | 11 | 2.59 | 3 | 0.65 | 0.40 | 3.27 | 2.50 | 0.93 | 0.68 |
| 4 | 8 | 2.50 | 2.50 | 0.51 | 0.42 | 3.35 | 3.13 | 0.72 | 0.29 |
| 5 | 10 | 4.80 | 4 | 0.42 | 0.20 | 4.66 | 3.90 | 0.57 | 0.36 |
| 6 | 16 | 2.69 | 3 | 0.48 | 0.23 | 2.94 | 3.00 | 0.51 | 0.26 |
| 7 | 4 | 3.50 | 4 | 0.37 | 0.40 | 3.37 | 3.40 | 0.51 | 0.42 |

Table 3: BRE statistical analysis for all stories.

| | | N | Mean Rank | Sum of Ranks | Z | P-value (2-tailed) | Cliff's Delta |
|---|---|---|---|---|---|---|---|
| BRE planning poker – BRE statistical combination | Negative Ranks | 28 | 22.48 | 629.50 | -1.890 | .059 | 0.171 |
| | Positive Ranks | 15 | 21.10 | 316.50 | | | |
| | Ties | 6 | | | | | |
| | Total | 49 | | | | | |

## 6 Lessons learned and recommendations

### 6.1 Preparatory activities

The preparatory activities were performed correctly, with the exception of release planning.

### 6.1.1 The Product Owner role

It was important that the Product Owner role be assigned to a person high enough in the management hierarchy to communicate its vision, as well as knowledgeable enough about the required functionality to maintain the Product Backlog and evaluate the implementation of user stories. The Product Owner independently managed the Product Backlog, defined priority of user stories, and wrote acceptance tests. In case of ambiguity, he promptly provided detailed explanations. He participated in all planning and review meetings, communicating the vision of what had to be developed and evaluating the work performed. Throughout the project, the mutual understanding and trust between the Product Owner and the Team increased and the amount of common knowledge grew.

### 6.1.2 The ScrumMaster role

From the viewpoint of professional qualifications, the role of ScrumMaster was also properly assigned. The problem was that the head of the web development department who played this role was too busy to devote enough time to the project. His deputy did not have computer science background, but was a media expert, which enabled her to strengthen the communication between the Product Owner and the Team. This had a positive effect on the project progress from the viewpoint of clarification of user requirements, but prevented her from performing more technically oriented ScrumMaster tasks. There was not enough attention paid to project planning and monitoring through burndown charts. Some deficiencies regarding story decomposition were noticed too late, which resulted in an unrealistic picture of the project progress shown by the Sprint burndown charts. These deficiencies could be prevented by providing the ScrumMaster with more training and experience or hiring the expert for this role during the first Scrum project and ensuring the transfer of knowledge.

### 6.1.3 Team composition

The companies that plan to implement the Scrum method should be careful about Team composition, in particular concerning coordination between members who cover different areas. All members must accept and obey Scrum rules. Adding new members in the middle of the project can cause disruptions and decrease the Team’s velocity. If this cannot be avoided, care should be taken that the new members are prepared to take the agile approach and are appropriately trained.

### 6.1.4 Concept of “done”

A common definition of “done” contributed significantly to the quality of the product. The Product Owner had an opportunity to promptly check the functionality and make sure that the necessary adjustments are made in time, thus avoiding comprehensive integration testing before the release delivery. It was important that the definition was established from the very beginning.

## 6.2 Progress monitoring

Progress monitoring was inadequate and deserves more attention in future. Although the actual velocity was measured systematically, the results were not considered appropriately during Sprint planning. Release planning was completely omitted and burndown charts were not used.

### 6.2.1 Velocity tracking

The initial velocity estimate was almost accurate, but later estimates were too optimistic. The velocity analysis revealed two recommendations for future Scrum projects: (1) the planned velocity should be estimated considering the actual velocity of previous Sprints instead of being the result of wishes and pressure, and (2) the development team should be as homogeneous and stable as possible. Overoptimistic velocity estimates lead to wrong expectations and a false sense of finishing on time instead of giving a realistic picture of progress.

### 6.2.2 Release planning

The lack of release planning was an important deficiency. A possible explanation for the absence of the release plan could be that the Team was “pushed” to the new portal development too early, before the initial Product Backlog was completely specified and a clear vision of the contents of the first release was known. Consequently, a great deal of requirements were specified later on, when the project was already well under the way. In order to manage such a situation, the Team should promptly maintain the release plan and monitor its implementation using release and Sprint burndown charts.

### 6.2.3 Sprint burndown charts

Apart from insufficient use of Sprint burndown charts, the major issue regarding Sprint progress monitoring was deficient story decomposition. Strictly following Scrum, the stories should be (at least roughly) decomposed into constituent tasks during the second part of the Sprint planning meeting. Late story decomposition was the reason that the Sprint burndown diagrams did not always present the actual project status, but sudden jumps/increases in the work remaining appeared even late in the Sprint.

### 6.3 Estimation accuracy

The planning poker method for story estimation was well accepted by the Team. The BRE analysis showed that the planning poker estimates were slightly more accurate than the statistical combination of individual estimates. However, the effect size measured with Cliff’s delta was small and the Wilcoxon signed-rank test did not confirm a statistically significant difference (*p*-value 0.059).

## 7 Conclusions

This paper describes a media portal development project with emphasis on issues affecting successful Scrum adoption. It focuses on three aspects: evaluation of the preparatory activities, critical analysis of the progress monitoring methods used, and analysis of the accuracy of effort estimates.

Most preparatory activities in the project studied were performed correctly. However, insufficient attention was given to the preparation of the initial Product

Backlog and the release plan. The planned velocity estimates tended to be too optimistic and were not based on actual achievement in previous Sprints. The BRE analysis showed that the planning poker estimates were slightly more accurate than the statistical combination of the individual estimates.

The study results can help the company studied to remove the deficiencies found and improve Scrum usage. The companies that plan to introduce Scrum into their development process can use the lessons learned to gain better insight into the Scrum adoption. The estimation accuracy analysis contributes to planning poker method research and determination of its key success factors.

## References

[1] Cliff, N., *Ordinal Methods for Behavioral Data Analysis*. Lawrence Erlbaum Associates, Mahwah, NJ, 1996.

[2] Grenning, J., Planning poker or how to avoid analysis paralysis while release planning, 2002. Online.
http://www.renaissancesoftware.net/files/articles/PlanningPoker-v1.1.pdf

[3] Hummel, M., State-of-the-art: A systematic literature review on agile information systems development. *Proc. of the 47th Hawaii Conference on System Sciences*, pp. 4712–4721, 2014.

[4] Jørgensen, M., Practical guidelines for expert-judgment-based software effort estimation. *IEEE Software*, Vol. 22, No. 3, pp. 57–63, 2005.

[5] Lavazza, L., Morasca, S., Taibi, D., Tosi, D., Applying SCRUM in an OSS development process: an empirical evaluation. *Lecture Notes in Business Information Processing*, Vol. 48, pp. 147–159, 2010.

[6] Mahnic, V., A case study on agile estimating and planning using Scrum. *Elektronika ir Elektrotechnika (Electronics and Electrical Engineering)*, Vol. 5, No. 111, pp. 123–128, 2011.

[7] Mahnic, V., A capstone course on agile software development using Scrum. *IEEE Transactions on Education*, Vol. 55, No. 1, pp. 99–106, 2012.

[8] Mahnic, V., Hovelja T., On using planning poker for estimating user stories. *Journal of Systems and Software*, Vol. 85, No. 9, 2086–2095, 2012.

[9] Moløkken-Østvold, K., Haugen, N. C., Benestad, H. C., Using planning poker for combining expert estimates in software projects. *Journal of Systems and Software*, Vol. 81, No. 12, pp. 2106–2117, 2008.

[10]Rising, L., Janoff, N. S., The Scrum software development process for small teams. *IEEE Software*, Vol. 17, No. 4, pp. 26–32, 2000.

[11] Schwaber, K., *Agile Project Management with Scrum*. Microsoft Press, Redmond, WA, 2004.

[12] VersionOne (2014). 8th Annual State of Agile Survey. Online. http://stateofagile.versionone.com/

[13]Wilcoxon, F., Individual comparisons by ranking methods. *Biometrics Bulletin*, Vol. 1, No. 6, pp. 80–83, 1945.

[14]Williams, L., Agile software development methodologies and practices. *Advances in Computers*, Vol. 80, pp. 1–44, 2010.