DOI: 10.2478/v10051-010-0005-2

Teaching Scrum in Cooperation with a Software Development Company

Viljan Mahnič[1], Strahil Georgiev[2], Tomo Jarc[3]

[1]University of Ljubljana, Faculty of Computer and Information Science, Tržaška 25, Ljubljana, viljan.mahnic@fri.uni-lj.si
[2]SRC Sistemske integracije, d.o.o., Tržaška 116, Ljubljana strahil.georgiev@src.si
[3]University of Ljubljana, IT department, Kongresni trg 12, Ljubljana, tomo.jarc@uni-lj.si

The increasing use of agile methods for software development creates the need for these methods to become part of the education of future computer and information science engineers. On the other hand, teaching these methods gives us an opportunity to verify individual agile concepts and their effectiveness. For that reason, project work is an appropriate and frequently used form of teaching that enables students to get acquainted with agile methods and, at the same time, provides case studies for evaluating individual agile concepts. We describe our approach to teaching the Scrum agile method, within the software technology course, in cooperation with a software development company. Students were taught through work on a real project for which a list of requirements was submitted by the company. A co-worker of this company participated throughout the teaching period playing the role of customer's representative. During their work, students consistently used the Scrum method and at the end of each iteration they evaluated their experience by means of a questionnaire. In the article, the Scrum method is presented first, then a description of work on the project is given and finally the results of the survey are described.

Key words: agile methods, Scrum, software development, computer engineering education, university-industry co-operation

1 Introduction

The role of agile methods (Abrahamsson et al., 2002) in software development is increasing. The results of the survey published by Dr. Dobb's Journal in 2008 (Ambler, 2008) show that the introduction of agile methods increases productivity, quality and satisfaction of the software development stakeholders. A comparison between companies that use the agile approach to software development and companies that use traditional disciplined approach (Ceschi et al., 2005) showed that adopting the agile approach improves project management and customer relationships. Nevertheless, in spite of the positive experience, there are still doubts about the effectiveness of agile methods, as if by using a few typical practices in extreme measure they bring instability and increase risk.

It is very important for the future computer science engineers to receive knowledge about agile methods during their studies. Since there are often opposite opinions about the effectiveness of the agile approach, teaching these methods offers many opportunities to check individual concepts in practice. Therefore, courses that deal with software development often include practical project work on almost real problems, so that students can familiarize themselves with the advantages and disadvantages of the agile approach. The examples of such courses include teaching of extreme programming (Shukla and Williams, 2002; Dubinsky and Hazzan, 2003), testing the effectiveness of test driven development and pair programming (Xu and Rajlich, 2006), and teaching the differences between agile and disciplinary approach to software development (Robillard and Dulipovici, 2008).

This was also the approach that we have chosen in the academic year 2008/09 for the final software technology course at the Faculty of Computer and Information Science at the University of Ljubljana. The course is taught to Computer Science students in the last (eighth) semester of their studies. We have selected the Scrum agile method (Schwaber, 2004), since it is one of the most widespread agile methods. Our goal was to test this method in circumstances which reflect reality and to monitor development process performance by using metrics, defined in (Mahnič and Vrana, 2007).

In order to enable students to work on an almost real project we have contacted the company SRC, one of the leading Slovenian software development companies. For the purpose of our course, SRC presented the students with the requirement specifications for the project "General Hospital Information System" and also provided an employee (a post-graduate student) who played the role of *Product Owner*. In Scrum terminology this is the customer's representative or domain expert responsible for functionality of the new software. Students' task was to implement the given requirements by using the Scrum method and at the same time provide measurements used for calculating the indicators of the development process performance.

In the next section we shall briefly present the main features of agile methods and give a short description of Scrum. In the third section we shall present in detail the students' project which was used as a case study for teaching Scrum and implementation of measurements used for performance monitoring and building the repository in line with CMMI requirements (Mahnič and Žabkar, 2007). The fourth section will be dedicated to the results of the survey for the analysis of students' satisfaction with the Scrum method and project work. At the end, we shall state the most important results and experience gained by the described approach.

2 Agile methods and Scrum

2.1 The main features of agile methods

Agile methods have emerged as an alternative to the traditional, heavily documented and disciplined approach to software development. The features of agile methods are simplicity, little documentation and fast response to changes requested by the user. At the same time, the user is more actively involved in development of the new product.

The foundations of agile movement were established in 2001, when a group of 17 consultants and practitioners gathered and published the four basic values of agile methods (Manifesto, 2001):

- individuals and interactions over processes and tools,
- working software over comprehensive documentation,
- customer collaboration over contract negotiation, and
- responding to change over following a plan.

Since then the usage of agile methods has been constantly increasing. The aforementioned survey (Ambler, 2008), in which 624 information technology experts have taken part (71% from North America, 17% from Europe and 4.5% from Asia), shows that 69% of the participants have worked on the projects managed by agile methodology. We can also see that the success rate of these projects is 77.5%, which is much higher than the traditional approach.

In the literature we can find many agile methods, such as:

- XP-Extreme Programming (Beck, 2000),
- FDD-Feature-Driven Development,
- Crystal,
- Scrum etc.

According to data referenced by Schwaber, Leganza and D'Silva (2007), the most popular agile methods are Extreme Programming and Scrum.

2.2 The Scrum method

The Scrum method emerged in the first half of the 1990s. The origin of its name is in rugby and means bringing the ball back into the game. It is the software development approach which directs as iterative and incremental way of work.

The project is divided into iterations named Sprints. Each iteration (Sprint) takes 30 calendar days and must result in the working software code which presents a new (additional) software functionality. Software code must be completely tested so that the customer can use it. In this way, the customer gradually receives individual parts of the solution that he/she can

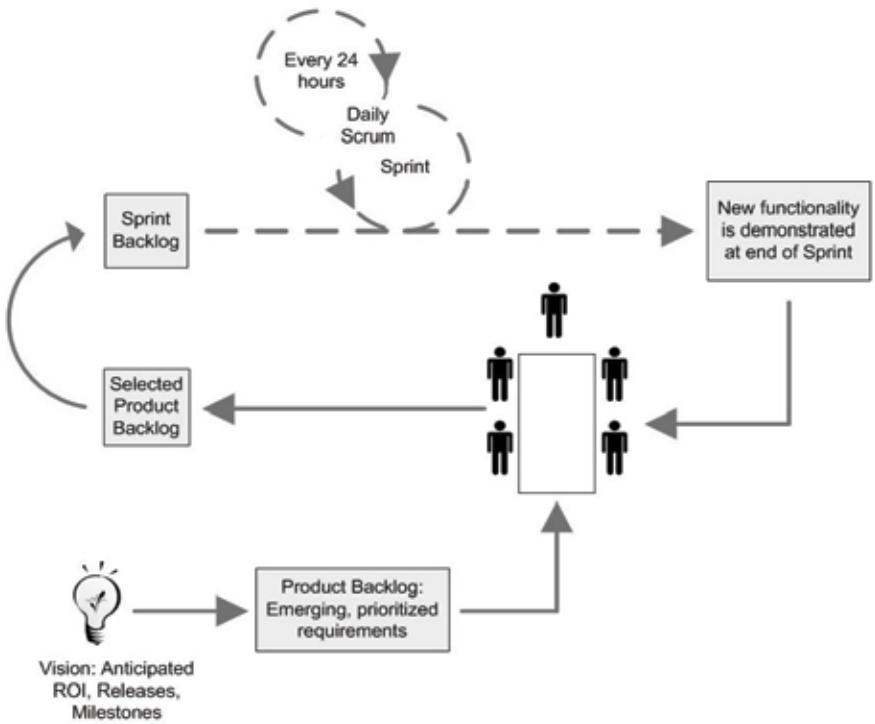


Figure 1: Scrum software development process (Schwaber, 2004)

immediately use. The Scrum method development process is shown in Figure 1.

The development is based on the list of the customer's requirements named Product Backlog. This list is maintained by the customer's representative (Product Owner) who represents users' needs and takes care of project financing. He/she adds new requirements if necessary and sorts them by priority determined by the present users' needs.

At the beginning of each iteration the customer's representative (Product Owner) meets with the development team (Team) so that they can together determine the subset of requirements to be developed in the next iteration. The meeting (Sprint Planning Meeting) takes 8 hours and consists of two parts, 4 hours each. In the first part the Product Owner and the Team agree which requirements from the list will be included in the next iteration. In the second part the development team builds the list of tasks necessary for implementation of the agreed requirements (Sprint Backlog). During the Sprint the bigger tasks are further divided, so that implementation of each task takes 4 to 16 hours of work.

During the iteration team members meet every day at a short 15-minute meeting (Daily Scrum Meeting) where each member answers three questions:

- What did you do yesterday?
- What are you planning to do today?
- What impediments stand in the way of meeting your commitments to this *Sprint* and this project?

In this way the project progress is transparent and immediate actions are taken when necessary.

At the end of each iteration the development team presents the results of its work to the *Product Owner* and all interested users. The presentation takes place at a special meeting named *Sprint Review Meeting* which enables users to comment on the work done and give their suggestions for the requirements to be developed in the next iteration.

Before the next iteration the development team meets with *ScrumMaster* (a person responsible for the Scrum process) in order to assess the work in the previous iteration and agree on the improvements that would increase performance and software quality in the next iteration.

Id	Product Backlog Item Description	Priority	Initial Estimate	Adjustment Factor	Adjusted Estimate
1	Creating new electronic medical record	1	40	1.3	52
2	Displaying electronic medical record	1	32	1.2	38
3	Searching electronic medical record	1	32	1	32
4	Changing electronic medical record	1	16	1	16
5	Dialog box for patient appointment reservation	2	40	1.3	52
6	Choosing a doctor	2	24	1	24
7	Displaying free terms of the doctor	2	24	1.2	29
8	Displaying the type of patient's appointment	2	24	1	24
9	Applying for an appointment	2	40	1.2	48
Sprint 1			**272**		**315**
10	Connection to the insurance company	2	24	1.3	31
11	Inserting an anamnesis	2	8	1	8
12	Displaying a list of medicaments	2	16	1	16
13	Displaying a list of diagnostics	2	16	1	16
14	Displaying a short CV of the doctor	2	8	1	8
15	Displaying a list of patients for an appointment	2	16	1	16
16	Inserting a new medical examination	2	16	1.5	24
17	Personnel records	2	32	1.2	39
18	Updating the status of the patient appointment and showing data for current patient	2	16	1.3	21
19	Operative interventions	2	40	1.3	52
20	Printing a prescription	3	24	1	24
21	Printing a medical certificate	3	24	1	24
22	Printing a doctor's note	3	24	1	24
23	Project documentation	3	20	1	20
Sprint 2			**284**		**323**
Release 1			**556**		**638**

Figure 2: List of requirements (Product Backlog)

The role of *ScrumMaster* is to a certain extent similar to the role of the project manager, but instead of determining and delegating individual tasks, he makes sure that the Scrum method is being followed in the most effective way. His important role is to provide the development team with optimal working conditions and to take care of immediate problem solutions.

The *Team* responsible for implementation of the required functionality has an interdisciplinary structure and is self organized. Team members delegate tasks by themselves and are collectively responsible for the success or failure of the project.

It is stated at the Scrum Community Wiki web page (2009) that Scrum is used by the biggest world companies, such as IBM, Microsoft, Oracle, Yahoo, Google, Toyota, BMW, etc. as well as many small and medium sized companies. Scrum is used for all types of projects including financial, web and health-care projects.

3 Student project case study

The aim of the student project at the final software technology course was to teach students to use the Scrum method on an almost real project based on real requirements of a specified customer. In order to find this kind of project the SRC company was contacted, which prepared the list of requirements and provided an employee who played the role of customer's representative (*Product Owner*).

Task ID	Task Description	Respon-sible	Task Status	Hours Spent					Hours Remaining					
1	2	3	4	1	2	3	...	13	0	1	2	3	...	13
T-2-01	Updating the data model	Patrik	Completed	5	2	0	...	0	10	5	1	1	...	0
T-2-02	Connection to the insurance company	Patrik	Completed	0	0	10	...	2	10	10	10	0	...	0
T-2-03	Inserting a type of insurance for a patient	Patrik	Completed	0	0	5	...	2	5	5	5	0	...	0
T-2-04	Displaying a list of medicaments	Rok	In Progress	0	0	0	...	2	10	10	10	10	...	0
T-2-05	Displaying a short CV of the doctor	Jure	Completed	0	0	0	...	0	5	5	5	5	...	0
T-2-06	Interface for the appointment list of the chosen date	Matevž	Completed	0	0	0	...	0	15	15	15	15	...	0
T-2-07	Updating the status of the patient appointment	Rok	Completed	2	0	2	...	1	10	8	8	6	...	0
T-2-08	Interface for measurements	Rok	Completed	0	0	0	...	1	10	10	10	10	...	0
T-2-09	Interface for diagnostics	Rok	Completed	0	0	0	...	0	10	10	10	10	...	0
T-2-10	Inserting operative interventions	Patrik	In Progress	0	0	0	...	0	15	15	15	15	...	0
T-2-11	Updating the status of the operative interventions	Patrik	In Progress	0	0	0	...	0	15	15	15	15	...	0
T-2-12	Printing a prescription	Matevž	Completed	0	0	0	...	2	10	10	10	10	...	0
T-2-13	Printing a medical certificate	Matevž	Completed	0	0	0	...	2	10	10	10	10	...	0
T-2-14	Printing a doctor's note	Matevž	Completed	0	0	0	...	0	10	10	10	10	...	0
T-2-15	Administrative interface for adding new doctors	Patrik	Completed	0	0	0	...	0	15	15	15	15	...	0
T-2-16	Administrative interface for adding new medical technicians	Patrik	Completed	0	0	0	...	0	15	15	15	15	...	0
T-2-17	Administrative interface for managing the scheduler	Patrik	Completed	0	0	0	...	0	15	15	15	15	...	0
T-2-18	SQL class<->data planning	Matevž	Completed	1	0	2	...	0	10	9	9	7	...	0
T-2-19	Authentication and connection with insurance database	Matevž	Completed	1	4	0	...	0	10	9	8	8	...	0
T-2-20	Implementation of control classes	Matevž	Completed	0	0	0	...	0	10	10	10	10	...	0
T-2-21	Updating the WEB interface	Jure	Completed	0	0	2	...	2	10	10	10	8	...	0
T-2-22	Updating the status of the patient appointment	Jure	Completed	0	0	0	...	0	10	10	10	10	...	0
Total				**9**	**6**	**21**	...	**12**	**240**	**231**	**226**	**205**	**...**	**0**
									240	**222**	**203**	**185**	**...**	**0**

Figure 3: Example of the Sprint Backlog form

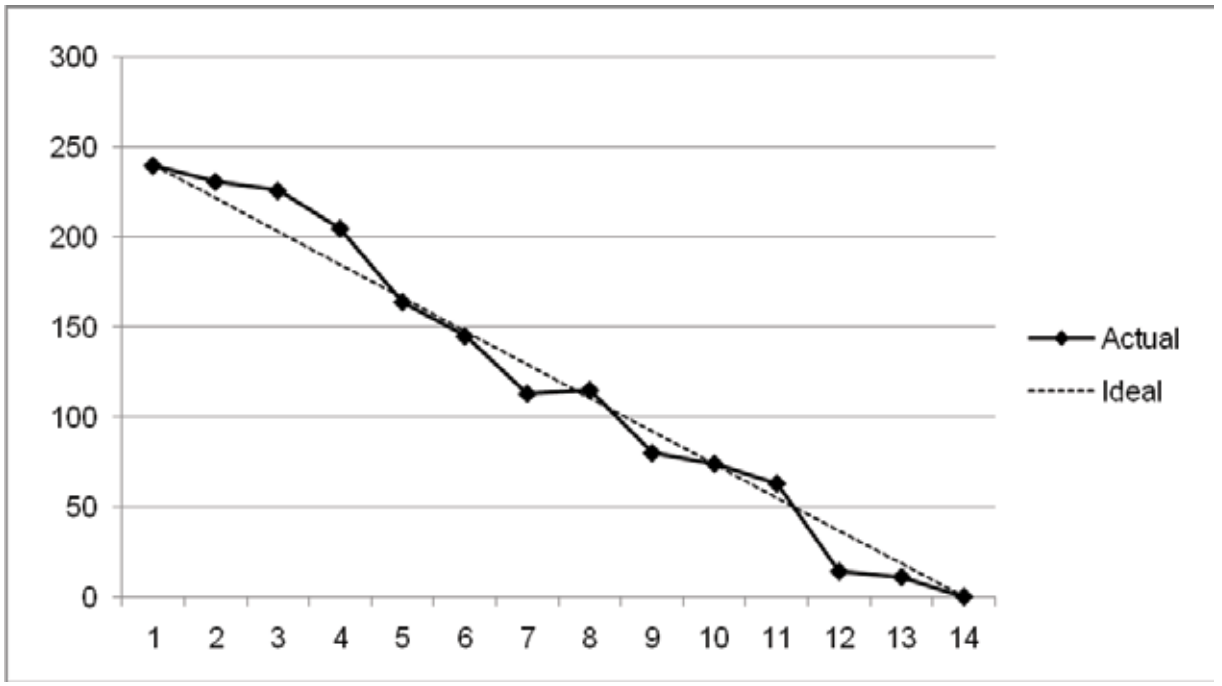


Figure 4: Graphical presentation of the amount of remaining work (in hours) – Sprint Burndown Chart

SRC is one of the leading Slovenian companies in the area of IT technologies and since its beginning has been supporting new ideas and project management methods with the aim of improving the internal working environment, quality of work and customer satisfaction. They have already implemented several projects using agile methods and intend to gain more experience and knowledge in this area. Therefore, the company has accepted the offer from the Faculty of Computer and Information Science in order to improve its knowledge about the theoretical and practical background of the Scrum method. On the other hand, the students have gained the opportunity to use the method on a real project provided by the company.

Since SRC and its company Infonet Kranj, d.o.o. have been offering solutions in the health area for a long time, the project was related to the development of the information system of a general hospital. The SRC company played the role of the customer, represented by their employee as a *Product Owner*. In order to make sure that the development would be in line with the Scrum method, we have precisely defined other roles on the project: the teacher played the role of *ScrumMaster* and students were grouped in three teams with four members. Each team independently developed the required software.

At the beginning, the *Product Owner* prepared the list of requirements (*Product Backlog*) shown in Figure 2. The requirements were grouped in several modules which included preparing and maintaining electronic medical records for each patient, patient appointment reservations and medical examination management, connection to the insurance company which provided personal data about patients and their insurance, and recording data on operative interventions. He also prepared a rough data model and code tables, such as the code table of medicaments.

The project has been divided into two iterations. As required by Scrum, each iteration started with the *Sprint Planning Meeting,* at which the *Product Owner* presented requirements, and ended with the *Sprint Review Meeting* at which development teams have presented the results of their work. At the end of each iteration we have organized a *Sprint Retrospective Meeting* at which we analyzed advantages and disadvantages in the previous *Sprint* and agreed on the improvements in the next iteration.

Because of the obligations that students had with other courses it was impossible to expect that *Daily Scrum Meetings* would take place every day, as requested by Scrum. In order to follow the Scrum method requirements as closely as possible we have asked students to have meetings twice a week: on Mondays and on Thursdays. On Mondays the meetings took place during lab hours, at which the teacher (as *ScrumMaster*) and SRC employee (as *Product Owner*) were present. On Thursdays the students had meetings on their own. There were 11 meetings during the first iteration, which lasted from 2nd March 2009 until 6th April 2009 and 13 meetings during the second iteration, which lasted from 9th April 2009 until 1st June 2009.

For each iteration every development team maintained its own task list (*Sprint Backlog*). For each task the team determined the team member responsible for the implementation and estimated the number of remaining working hours necessary for the task implementation. At the *Daily Scrum Meeting* students recorded the number of hours spent on each task and estimated the number of hours remaining until completion of the task. The Scrum method requires only recording the

amount of work remaining, but recording the amount of work spent also enabled us to monitor performance indicators of the development process in the model described in (Mahnič and Vrana, 2007) and (Mahnič and Žabkar, 2007). In this way the project presented also the case study for the implementation of this model.

We have prepared a special form for maintaining the *Sprint Backlog*. The completed form for one of the development groups is shown in Figure 3. Students sent the filled form to *ScrumMaster* after each *Daily Scrum Meeting*. The data about hours spent and remaining enabled the *ScrumMaster* and development team to regularly monitor the work progress.

The total amount of work remaining was shown after each *Daily Scrum Meeting* as a chart named *Sprint Burndown Chart* which enabled comparison between the actual project progress and an ideal situation with the amount of work remaining decreasing linearly across time. *Sprint Burndown Chart* for the task list in Figure 3 is shown in Figure 4.

4 Questionnaire Analysis

After each iteration the students were asked to answer a questionnaire in order to get the response on their satisfaction with the project progress and their opinion on the Scrum method. 30 students participated – besides the students working on the hospital information system also the students working on the tool for project management based on Scrum. The questionnaire had 14 questions, for each question answers ranged from 1 to 5. Grade 1 was the worst and grade 5 was the best. For each question the students could write their comments and explain the grade.

4.1 List of Requirements

The first two questions were related to the list of requirements *(Product Backlog*).

Question 1: **Clarity of initial *Product Backlog* (Was the *Product Backlog* for the current *Sprint* clearly determined? Did you understand the *Product Owner* requirements from the short description for each requirement?)**

The general response was that the description of individual requirements was too short and not specific enough. But the majority of questions were answered at the meetings where *Product Owner* participated. As shown in Table 1 the average grade for this question improved significantly after the second iteration. The reason might be that we have prepared additional user cases for both projects, which gave students a better understanding of the requirements.

Question 2: **Time estimate for the individual requirements from *Product Backlog* (Were the time estimates for the working hours required appropriate?)**

The majority of students answered that the initial estimates agreed with the *Product Owner* were correct. The grade for this question also improved significantly in the second iteration.

4.2 The task list maintenance

Question 3: **Administration of the Scrum method (Were the spreadsheets clear and easy to understand?)**

Question 4: **Administration workload**

Maintaining the task list (*Sprint Backlog*) and recording the number of hours spent and remaining required additional administrative work from the members of the development team. Therefore, we were interested to find out how students evaluate this additional workload. The answers have shown that the students had problems at the beginning, because the procedure of filling the *Sprint Backlog* form was not clear, especially for the cases when bigger tasks had to be split into smaller ones and the initial estimate of work remaining had to be replaced with estimates for the new tasks. But later the students got used to the principles of entering data so that there were no special problems. This is reflected in the average grade shown in table 2 which rose in the second iteration from 3.7 to 4.3. Regarding question 4 we can see from the average grade that the students were equally satisfied with the administration workload, since the average grade 3.3 did not change.

Table 1: Average grades for the questions related to the list of requirements

Question	Sprint 1	Sprint 2
Clarity of initial Product Backlog	3.2	3.9
Time estimate for the individual requirements from Product Backlog	3	3.8

Table 2: Average grades for the questions related to the Sprint Backlog maintenance

Question	Sprint 1	Sprint 2
Administration of Scrum method	3.7	4.3
Administration workload	3.3	3.3

4.3 Technical and content problems

Question 5: **Technical problems at the beginning of the *Sprint***

Question 6: **Technical problems at the end of the *Sprint***

Members of each development team made the choice of development technology by themselves. A few groups selected familiar technologies, already used by some members of the team. Some groups decided to use new technologies and wanted to gain additional experience and knowledge, so they had more problems at the start. Technical problems were also related to integration of code written by different developers.

We can see from the answers that in the first iteration there were more problems at the beginning of the *Sprint* (average grade 3.3) and fewer at the end (average grade 3.9). By contrast, in the second iteration there were fewer technical problems at the beginning of the *Sprint* (average grade 4.1) and more at the end (average grade 3.7). This can be explained by the fact that at the beginning of the second iteration the students had already established the required technical infrastructure, but were coping with integration of the code into operational solution at the end. The details are shown in Table 3.

Question 7: **Content problems (understanding required functionality) at the beginning of the *Sprint***

Question 8: **Content problems (understanding required functionality) at the end of the *Sprint***

Regarding the content problems it was important that the development teams had no user representative who could promptly answer the developers' questions. Even though the Scrum method demands an interdisciplinary development team (including the user representatives), we could not organize it since all team members were developers. Therefore, the students suggested that it would be better if the customer's representative would test the software during the iteration and give comments promptly (and not at the end). In the first iteration the average grade for question number seven was 3.5 and for the eighth question 4.1. Similarly to the questions related to the technical problems, we can see that the content problems increased at the end of the second iteration, when individual programs had to be integrated in the operational solution. The average grades are shown in Table 3.

4.4 Cooperation with other project stakeholders

Question 9: **Scrum Master Cooperation**

Question 10: **Product Owner Cooperation**

Question 11: **Cooperation with other team members**

Regarding questions number 9 and 10 the students were satisfied with the *ScrumMaster* and *Product Owner* cooperation. Regarding question 11 many students made comments that they knew each other very well from before and this made their working together easier. With more heterogene-

Table 3: Average grades for questions about technical and content problems

Question	Sprint 1	Sprint 2
Technical problems at the beginning of the *Sprint*	3.3	4.1
Technical problems at the end of the *Sprint*	3.9	3.7
Content problems at the beginning of the *Sprint*	3.5	3.8
Content problems at the end of the *Sprint*	4.1	3.7

Table 4: Average grades for the questions about cooperation with other project stakeholders

Question	Sprint 1	Sprint 2
Cooperation with *ScrumMaster*	4	4.3
Cooperation with *Product Owner*	3.8	4
Cooperation with other team members	4	4.1

Table 5: Average grades for general questions

Question	Sprint 1	Sprint 2
Appropriateness of the scope of project work	3.8	3.7
General estimate of satisfaction with project work	3.7	3.8
General estimate of the Scrum method	3.8	3.9

ous groups there were more problems in this area. All grades improved in the second iteration, which shows that the Scrum method positively affects relationships and teamwork. Average grades for each question can be seen in Table 4.

4.5 General questions

Question 12: **Appropriateness of the scope of project work (Was the scope of project work appropriate?)**

Question 13: **General estimate of satisfaction with project work**

Question 14: **General estimate of the Scrum method (Was this method useful for the development team? Would you recommend it to other developers?)**

The answers to question number 12 show that the scope of the project work was appropriate, so that the majority of students were not overloaded and they could fulfill other student obligations at the faculty. The students were rather satisfied with the project progress and the method. We can see from their comments that they consider the Scrum method appropriate for work in bigger teams and on bigger projects. Their opinion was that the method importantly increases the transparency of development progress without demanding a lot of administration, which is difficult for the developers. Average grades for this group of questions are shown in table 5.

5 Conclusion

The approach to teaching the final software technology course described in this paper represents a continuation of our efforts to ensure closer cooperation with software companies, already presented in one of our previous papers (Mahnič, 2008). The experience has shown that this kind of co-operation benefits everyone involved in the pedagogical process.

While working on a real project the students obtained knowledge of the advantages and disadvantages of the Scrum method, and were introduced to the problem of quantitative monitoring of the development process, which is an important research challenge for agile methods. They also gained practical experience and increased their transferrable skills like teamwork, communication, planning and task delegating, presenting the solution etc. This kind of knowledge cannot be communicated through formal lectures, but only in a professional working environment.

The involvement of SRC in teaching this course enabled the company to test one of the potentially interesting agile methods without risk and additional workload for its employees, so that it could use that method in its operations. The SRC employee who was involved in the project could estimate the advantages and disadvantages of Scrum on the basis of experience and could find the way of implementing this method in the regular procedure of the company. In this way we have transferred the knowledge from the academic world to the practice, which does not happen as often as we hope and need. Based on the practical experience gained, SRC will improve its internal method of software development.

Co-operation with industry enabled the teacher to expose students to one of the agile methods in a practical way. The experience has shown that students' learning motivation increases if they can test their knowledge in practice. At the same time this project had an important research component: it was used as a case study for evaluation of the measurement model developed at the faculty. This project helped us to gather the real data necessary for calculating the performance indicators for software development using the earned value method.

References:

Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. (2002). *Agile software development methods,* VTT Electronic, Espoo.

Ambler, S. W. (2008). Has Agile Peaked? Let's look at the numbers, *Dr. Dobbs Journal*, May 2008, available from: http://www.ddj.com/architect/207600615 (9.6.2009).

Beck, K. (2000). *Extreme Programming Explained*, Addison-Wesley, 2000.

M. Ceschi et al. (2005). Project Management in Plan-Based and Agile Companies, IEEE *Software*, 22(3): 21-27.

Dubinsky, Y. & Hazzan, O. (2003). eXtreme Programming as a Framework for Student-Project Coaching in Computer Science Capstone Courses, *Proceedings of the IEEE International Conference on Software – Science, Technology & Engineering (SwSTE'03), Herzlia, Israel, November 4-5, 2003*.

Mahnič, V. (2008). Teaching Information System Technology in Partnership with IT Companies, *Organizacija*, 41(2): 71-78. DOI: 10.2478/v10051-008-0008-4

Mahnič, V. & Vrana, I. (2007). Using stakeholder driven process performance measurement for monitoring the performance of a Scrum based software development process, *Electrotechnical Review*, Ljubljana, 74(5): 241-247.

Mahnič, V. & Žabkar, N. (2007). Introducing CMMI Measurement and Analysis Practices into Scrum-based Software Development Process, *International Journal of Mathematics and Computers in Simulation*, 1(1): 65-72.

Manifesto for Agile Software Development, available from http://www.agilemanifesto.org/ (9.6.2009).

Robillard, P. N. & Dulipovici, M. (2008). Teaching Agile versus Disciplined Processes, *International Journal of Engineering Education*, 24(4): 671-680.

Schwaber, C ., Leganza, G. & D'Silva, D. (2007), The Truth About Agile Processes, available from: http://www.forrester.com/Research/Document/0,7211,41836,00.html (05.06.2009)

Schwaber, K. (2004). *Agile Project Management with Scrum*, Microsoft Press, Redmond.

Scrum Community Wiki (2009). Firms Using Scrum, available from: http://scrumcommunity.pbworks.com/Firms-Using-Scrum (05.06.2009)

Shukla, A. & Williams, L. (2002). Adapting Extreme Programming For A Core Software Engineering Course, *Proceedings of the 15th Conference on Software Engineering Education and Training (CSEET'02),* Covington, Kentucky, February 25-27, 2002.

Xu, S. & Rajlich, V. (2006). Empirical Validation of Test-Driven Pair Programming in Game Development, *Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse (ICIS-COMSAR'06*), *Honolulu, Hawaii, July 10-12, 2006.*

Viljan Mahnič is an Associate Professor and the Head of the Software Engineering Laboratory at the Faculty of Computer and Information Science of the University of Ljubljana. From 1999-2003 and from 2005-2006 he was also the Vice-Dean for educational affairs. He teaches courses in computer programming, software engineering and information systems technology at undergraduate and postgraduate levels. His research interests include software technology and development of information systems. He has been a member of the Board of Directors of EUNIS (European University Information Systems Association) since 2002 and the Vice-President since June 2009. He is a member of the IEEE Computer Society and the AIS (Association for Information Systems).

Strahil Georgiev graduated in Computer Science in the year of 2007, from the Faculty of Computer and Information Science, University of Ljubljana. In the same year he joined the SRC company as a software developer. Currently he is undertaking postgraduate studies in Information Systems and Decision Making at the same Faculty.

Tomo Jarc received his M.Sc. degree at the University of Ljubljana. He is the deputy of the IT department at the University of Ljubljana. Before that, he worked at SRC company as a consultant in the department of IT services and operations. He has a lot of experience in software development as well as in managing large software development and IT service support teams.