Assessing teamwork in a software engineering capstone course

Marko Poženel

University of Ljubljana
Ljubljana, Slovenia

ABSTRACT: The quality of teamwork within a software engineering capstone course requiring students to develop a quasi-real project following the Scrum agile software development methodology was analysed in this study. The analysis is based on an instrument recommended in the literature that addresses key characteristics of teamwork and presents them along five dimensions: shared leadership, team orientation, redundancy, learning and autonomy. The results of 33 student teams who attended the courses Software Engineering (Bachelor level) and Modern Methods of Software Development (Master level) in the academic year 2011/2012 are presented and compared to results from industry reported in the literature. The comparison revealed a fair degree of similarity between students and professional developers. Both rated the highest, learning and shared leadership, while autonomy received the lowest grade. However, professional developers were much more uniform in their opinions than were students.

INTRODUCTION

Over the past 15 years, agile methods have become popular in the field of software development. Forrester reported that in 2009 agile methods were used in more than 35% projects [1]. As of 2012, according to Gartner's forecasts, up to 80% of software development projects are conducted in the agile way [2]. User experience shows that the use of agile methods significantly increases the productivity, quality and satisfaction of all participants in the development process [3]. Although the prevailing opinion is that agile methods are appropriate mainly for small projects, the introduction of agile development methods is also reported in the largest companies in the field of information technology.

The increasing use of agile methods requires that they be introduced into computer science curricula [4]. Universities still place greater emphasis on traditional disciplined approaches and, until recently, all attempts to teach agile methods were regarded as a significant novelty [5]. At the Faculty of Computer and Information Science of the University of Ljubljana, Ljubljana, Slovenia, agile methods are taught in the context of the final capstone project, which students develop in the last semester of their studies. In this way, students also acquire practical skills that cannot be provided in a purely academic environment. At the same time, student projects serve as a case study for the use of agile methods in practice. To teach agile methods Scrum was chosen, which is considered to be the most widely used agile method [6]. According to the latest survey, Scrum is used by about 66% of agile projects [7].

In the academic year 2008/09, when the course was taught for the first time, students' perceptions and teacher's observations were analysed in order to improve the design of the course [8]. The results showed that the students were overwhelmingly satisfied. However, close monitoring of the student teams' performance revealed that students lacked the abilities to estimate and plan, and did not fully understand the Scrum concept of a task or user story being actually *done*. Therefore, the course was upgraded in the academic year 2009/2010, with the project work being designed as an observational study providing data for empirical evaluation of students' skills, with special emphasis on user stories estimation, release planning, fulfilment of the scope and velocity tracking [9][10]. The design of the course was presented as an example of good practice in teaching agile methods [11].

A survey on students' perceptions of Scrum practices revealed that students consider teamwork and good communication within the development team the most important factors affecting the success of a Scrum project [12]. Teamwork in agile projects is the subject of numerous studies. Effectiveness of Scrum teams was analysed by Moe and Dingsøyr [13]. In order to evaluate the quality of teamwork in agile projects, Moe et al developed a set of open-ended questions, which were used in interviews with the members of Scrum teams they studied in two Norwegian companies [14]. On the basis of these questions, Stettina and Heijstek prepared a questionnaire suitable for a survey, and empirically evaluated the quality of teamwork of eight development teams from eight different countries [15].

Using this questionnaire, the author surveyed the students who worked on their capstone projects within the courses Software Engineering (SE) and Modern Methods of Software Development (MMSD) in the academic year 2011/2012. The course SE is taught in the 3rd year of the new undergraduate Bologna study programme, while the course MMSD is an elective course at Master's level. In this article, the author has presented the survey results and compared them with the results published by Heijstek and Stettina [15]. In the rest of the article, the author has first described the teamwork assessment method and the questionnaire used. After that, the author has presented the survey results and the analysis of them. The main findings are summarised in the conclusion.

METHOD FOR TEAMWORK ASSESSMENT

To assess the students' teamwork, the author used the method proposed by Moe et al, which is based on a detailed analysis of the following five dimensions that are important for successful work on agile projects: shared leadership, team orientation, redundancy, learning and autonomy [14]:

- Shared leadership: In development teams who operate on the principle of self-organisation, leadership and decision-making has to be distributed among all members of the team. Team members make important decisions together. Discouraged are both the centralisation of decision-making, where one person decides on everything, and a complete decentralisation, where each member of the team decides on his work individually and independently of the others. It is important that the person who possesses the most knowledge, skills and abilities about the issues that are currently being solved takes the initiative. Consequently, the activities of the project manager have to be focused on creating and maintaining a vision, and on providing resources and confidence-building, while the members of the team should be left to take control when they have the necessary knowledge to be used or shared with each other in various stages of the project.
- Team orientation: Teamwork orientation is reflected in the fact that the team goals are given priority over individual goals. Salas et al showed that focusing on team goals increases the efficiency, dedication and satisfaction of individual team members and thus the productivity of the team as a whole [16]. They also reported on studies which found that teamwork-focused development teams more frequently devote attention to the opinions of other team members.
- Redundancy: In teams with a high degree of teamwork, individual members often can be replaced by other team members or help them in their work without a need for extensive additional training. Therefore, members of agile development teams, who work on the principle of co-operation and self-organisation, should not be narrowly specialised. Rather, they need to have multiple skills, so that they are able to perform other team members' jobs and substitute for another if circumstances require so. For efficient mutual assistance and replacement, it is also important that each team member is always familiar with the tasks of other members, so that the kind of help needed is clear at any given time.
- Learning: Learning is reflected in the fact that the development team is able to develop a common picture of the problems faced and, thus, is able to find solutions that improve team performance. Moreover, the agile development teams have to continually adapt to transformations arising from the wider environment. Successful teams tend to constantly improve their work methods on different levels.
- Autonomy: This dimension describes the extent to which a team is autonomous in its work. Team autonomy is defined as the influence of management and other individuals outside the team with the requests to take decisions regarding the strategy of work, processes, project goals, resource allocation, etc. External influence on the development team sometimes can be beneficial because it may assist in the completion of the project and increase creativity. However, Hoegl and Parboteeah consider the external impact on software-development teamwork as usually harmful [17].

To analyse each of these dimensions, Moe et al developed a set of open-ended questions, which were posed to all members of a development team within a 20-minute interview [14]. Based on the responses from team members, each dimension was evaluated with a score from 0 to 10, and the results were presented in the form of a radar plot. The radar plot was, then, used as a basis for a feedback session with the development team, where the results were analysed and possibilities for improvement were discussed.

Stettina and Heijstek [15] transformed the open-ended questions proposed by Moe et al [14] into quantitative questions and used them as the basis for an anonymous survey among eight development teams from eight different countries. The author adopted their questionnaire, with a few minor adjustments, for the survey among surveyed students, so that the results of both surveys are directly comparable. The answers to the questions were provided using a five-point Likert scale, where the items have the following meanings: 5 - Strongly Agree, 4 - Agree, 3 - Neutral, 2 - Disagree, 1 - Strongly Disagree. All questions have been asked in such a way that a higher score meant a better result, i.e. a better self-organised development team. The questionnaire is shown in Table 1.

The author was interested in the mean value and the consistency of the responses within each dimension group and in the mean value and the consistency of responses from all students. The consistency of the responses was estimated using the variance (σ^2) as presented in Equation 1:

$$\sigma^2 = \frac{\sum(x-\mu)^2}{N} \quad (1)$$

Where, N denotes the number of responses to an individual question, X denotes the answer value, and μ is the mean value. A lower variance, therefore, indicates a higher degree of agreement of answers within a team and vice versa.

Table 1: Questionnaire for the analysis of teamwork.

1. Shared leadership
Creation and maintenance of the team's shared mental model and transfer of leadership according to key knowledge, skills and abilities, shared decision authority.
• I feel everyone is involved in the decision-making process.
• I feel team members do not make important decisions without consulting other team members.
• I feel the team vision is well-defined and presented.
• I feel the team is designed (and redesigned) according to its purpose.
2. Team orientation
Promotion of team cohesion counteracts social loafing and increases individual responsibility, team goals are given priority over individual goals.
• I feel the team takes into account alternative suggestions in team discussions.
• I feel the team values alternative suggestions.
• I feel team members relate to the tasks of individuals.
• I regularly comment on a co-worker's work.
3. Redundancy
Cross-functionality avoids bottlenecks and enables the opportunity to shift workloads, and to assist mutually.
• I feel it is easy to complete someone else's task.
• I feel I get help if I get stuck.
• I help others when they have problems.
• I feel it is easy to substitute a person if someone leaves the team.
4. Learning
Interdisciplinary knowledge acquisition to promote self-optimisation in a wider environment.
• I feel the team keeps what works well in the development process.
• I feel the team improves the development method when software development problems are identified.
• I feel the team gives feedback on all aspects of each other's work.
5. Autonomy
External influences on the activities of the team, a precondition for self-management. Although sometimes beneficial, such influences can discourage group thinking.
• I feel the team does not lose too much time and resources to other projects.
• I feel people and groups outside the team do not have influence over important operational decisions in the project.
• I feel decisions made by the team are respected by people and groups outside the team.

RESULTS OF THE SURVEY

The survey was answered by 159 students out of 173 who attended both courses, thus, representing a response rate of 91.9%. Each course required students to work in groups in order to develop a quasi-real project on the basis of user requirements provided by a domain expert playing the role of the Product Owner. As a rule, a development team consisted of five students. Only in exceptional cases did a team consist of four or six students. The first three weeks served as a preparatory Sprint (also called Sprint 0) before the start of the project. During Sprint 0 formal lectures took place in order to teach students Scrum, and how to apply user stories for requirements specification and project planning. These three weeks were also used to prepare the development environment and acquaint students with the initial Product Backlog containing a set of prioritised user stories for the project they were going to develop. At the end of Sprint 0, each team estimated the stories using planning poker and prepared the release plan.

The rest of the course was divided into three Sprints, each lasting four weeks. Strictly following the Scrum method, each Sprint started with a Sprint planning meeting, and ended with the Sprint review and the Sprint retrospective meetings. During the Sprint, the teams have to meet at the Daily Scrum meetings twice per week and maintain their Sprint Backlogs. After three Sprints, the first release had to be complete and delivered to the customer.

Strictly following Scrum principles, the students acted as self-organising and self-managing Scrum Teams who were collectively responsible for the implementation of the required functionality. The instructors did not interfere in the distribution of tasks among team members and the estimation of effort, but played the role of Scrum Masters. They merely acted as facilitators, ensuring that Scrum ran smoothly and delivered the expected benefits.

The survey data were analysed both as a whole and by individual teams. Each response was considered valid if it contained answers to all questions. The results are gathered in Table 2. For each development team, the team size, the number of respondents, and the mean and variance for each dimension are presented. In the last column, the variance of all answers (regardless of dimension) is shown. The teams who attended the course MMSD are marked with the letter S, and the teams who attended the course SE are marked with the letter T. The radar diagram in Figure 1 shows the summarised results for each of the two courses and for all 159 respondents.

Table 2: Mean (x) and variance (σ^2) by teams for all 5 dimensions (min and max).

Development team	Team size	Number of respondents		Shared leadership	Team orientation	Redundancy	Learning	Autonomy	Average team agreement
S1	5	5	x	2.80	2.85	3.20	3.27	3.00	
			(σ^2)	*(0.76)*	*(0.83)*	*(1.26)*	*(0.86)*	*(1.6)*	*(1.06)*
S2	5	5	x	3.60	3.90	2.95	3.60	2.73	
			(σ^2)	*(0.84)*	*(1.29)*	*(1.85)*	*(0.51)*	*(1.93)*	*(1.28)*
S3	5	4	x	4.44	4.00	3.56	4.42	3.50	
			(σ^2)	*(0.62)*	*(0.88)*	*(1.5)*	*(0.24)*	*(2.25)*	*(1.1)*
S4	5	4	x	3.69	3.50	2.94	3.58	2.75	
			(σ^2)	*(0.34)*	*(0.5)*	*(1.31)*	*(0.41)*	*(1.35)*	*(0.78)*
S5	5	5	x	4.55	4.25	3.50	4.47	3.07	
			(σ^2)	*(0.35)*	*(0.69)*	*(2.15)*	*(0.25)*	*(1.8)*	*(1.05)*
S6	5	5	x	4.40	4.55	4.20	4.47	2.73	
			(σ^2)	*(0.34)*	*(0.35)*	*(1.06)*	*(0.25)*	*(1.4)*	*(0.68)*
S7	5	5	x	3.90	4.45	3.05	4.07	3.27	
			(σ^2)	*(0.99)*	*(0.55)*	*(1.75)*	*(0.46)*	*(0.33)*	*(0.82)*
S8	5	5	x	4.00	4.30	3.60	4.13	3.20	
			(σ^2)	*(0.4)*	*(0.31)*	*(2.14)*	*(0.38)*	*(0.96)*	*(0.84)*
S9	5	5	x	4.10	4.35	3.85	4.20	2.73	
			(σ^2)	*(0.19)*	*(0.43)*	*(1.13)*	*(0.29)*	*(1.53)*	*(0.71)*
S10	6	6	x	4.25	4.00	3.92	4.28	3.19	
			(σ^2)	*(0.35)*	*(0.58)*	*(0.91)*	*(0.31)*	*(1.4)*	*(0.71)*
S11	5	5	x	3.80	3.95	3.20	3.80	2.87	
			(σ^2)	*(0.56)*	*(0.85)*	*(1.36)*	*(0.43)*	*(1.45)*	*(0.93)*
S12	5	3	x	3.00	3.83	3.58	3.44	2.22	
			(σ^2)	*(0.67)*	*(1.14)*	*(0.74)*	*(0.91)*	*(1.51)*	*(0.99)*
S13	6	5	x	4.50	4.20	3.55	4.40	3.00	
			(σ^2)	*(0.25)*	*(0.76)*	*(1.35)*	*(0.37)*	*(1.47)*	*(0.84)*
S14	5	5	x	4.20	4.00	3.50	4.07	2.53	
			(σ^2)	*(0.56)*	*(0.6)*	*(1.55)*	*(0.6)*	*(1.58)*	*(0.98)*
S15	5	5	x	3.74	3.50	2.75	3.47	3.15	
			(σ^2)	*(0.61)*	*(0.75)*	*(1.29)*	*(0.38)*	*(1.36)*	*(0.88)*
S16	5	5	x	4.40	3.75	3.45	4.13	2.53	
			(σ^2)	*(0.44)*	*(1.09)*	*(1.65)*	*(0.25)*	*(2.38)*	*(1.16)*
S17	5	5	x	3.75	3.60	3.75	3.67	3.13	
			(σ^2)	*(0.19)*	*(0.44)*	*(1.29)*	*(0.62)*	*(1.45)*	*(0.8)*
S18	5	5	x	3.85	3.30	3.80	3.40	2.80	
			(σ^2)	*(0.73)*	*(0.81)*	*(1.06)*	*(0.77)*	*(1.63)*	*(1)*
S19	4	4	x	4.50	4.31	3.69	4.67	2.83	
			(σ^2)	*(0.38)*	*(0.59)*	*(1.84)*	*(0.22)*	*(1.97)*	*(1)*
S20	4	4	x	3.81	3.88	3.56 *(1)*	3.83	3.42	
			(σ^2)	*(0.53)*	*(0.61)*		*(0.64)*	*(1.24)*	*(0.8)*
S21	5	5	x	4.11	3.95	4.05	4.00	3.33	
			(σ^2)	*(0.52)*	*(0.45)*	*(0.85)*	*(0.4)*	*(0.76)*	*(0.59)*
S22	5	5	x	3.75	3.50	3.65	3.67	3.07	
			(σ^2)	*(0.79)*	*(0.85)*	*(1.73)*	*(0.36)*	*(1.26)*	*(1)*
S(MMSD) summary			x	3.96	3.91	3.51	3.96	2.96	
			(σ^2)	*(0.52)*	*(0.70)*	*(1.40)*	*(0.45)*	*(1.48)*	*(0.91)*

T1	4	4	x (σ^2)	4.19 *(0.28)*	3.56 *(0.62)*	3.25 *(2.31)*	4.33 *(0.39)*	3.33 *(0.89)*	*(0.9)*
T2	5	5	x (σ^2)	4.05 *(0.55)*	4.05 *(0.65)*	3.60 *(1.84)*	4.07 *(0.33)*	3.20 *(1.36)*	*(0.94)*
T3	5	5	x (σ^2)	3.90 *(1.19)*	4.00 *(0.4)*	3.90 *(1.39)*	4.47 *(0.65)*	3.07 *(0.73)*	*(0.87)*
T4	6	6	x (σ^2)	3.92 *(0.49)*	3.63 *(0.57)*	3.67 *(1.14)*	3.94 *(0.16)*	2.94 *(0.83)*	*(0.64)*
T5	5	5	x (σ^2)	3.30 *(1.41)*	3.45 *(0.75)*	3.00 *(1.4)*	3.20 *(1.76)*	3.13 *(1.85)*	*(1.43)*
T6	5	5	x (σ^2)	3.70 *(0.41)*	3.75 *(0.39)*	3.70 *(1.21)*	3.80 *(0.69)*	2.67 *(1.69)*	*(0.88)*
T7	5	5	x (σ^2)	4.30 *(1.11)*	3.95 *(0.95)*	3.45 *(1.15)*	4.33 *(0.76)*	3.29 *(1.49)*	*(1.09)*
T8	5	5	x (σ^2)	4.10 *(0.49)*	4.05 *(0.25)*	3.45 *(0.45)*	4.07 *(0.33)*	2.71 *(0.63)*	*(0.43)*
T9	5	5	x (σ^2)	4.25 *(0.79)*	3.85 *(0.93)*	3.40 *(1.04)*	4.13 *(0.25)*	3.33 *(0.89)*	*(0.78)*
T10	6	5	x (σ^2)	4.15 *(0.43)*	4.15 *(0.63)*	3.80 *(1.56)*	4.20 *(0.56)*	3.07 *(1.26)*	*(0.89)*
T11	5	4	x (σ^2)	4.06 *(0.43)*	4.00 *(0.5)*	4.19 *(0.4)*	3.92 *(0.58)*	3.42 *(1.58)*	*(0.7)*
T(SE) summary			x (σ^2)	3.99 *(0.69)*	3.86 *(0.60)*	3.58 *(1.26)*	4.04 *(0.59)*	3.11 *(1.20)*	*(0.87)*
Summary - all students			x (σ^2)	3.97 *(0.57)*	3.89 *(0.67)*	3.54 *(1.35)*	3.98 *(0.50)*	3.01 *(1.39)*	*(0.90)*

DISCUSSION

The students of both courses rated as lowest the autonomy dimension. Within the course SE, eight teams out of 11 (82%) assigned the smallest score to this dimension, while within the course MMSD, 19 teams out of 22 (86%) did so. The students justified their low estimates as a result of the workload for other courses and, as a consequence, the need to co-ordinate the work on the project with the rest of their workload. On the other hand, the learning and the shared leadership dimensions were rated highest. Within the course SE, seven teams out of 11 (64%) assigned the highest score to learning, two teams to shared leadership, and the remaining two teams to team orientation. Eleven MMSD teams assigned the highest score to shared leadership, eight teams to team orientation, and three teams to the learning dimension.

The author believes the main reason that most SE teams rated the learning dimension highest stems from the fact that they had previously attended the course Web Programming and had, thus, gained more knowledge necessary for the capstone project. Their programming knowledge enabled them to focus less on the technical part and more on the development process and its improvement. In contrast, the MMSD students (due to peculiarities of their previous education) had less programming experience and had to focus more on plain programming. The high score, that the MMSD students assigned to shared leadership is, in the author's opinion, more or less a result of the heterogeneity of the development teams (many students had not known each other before), which effectively led to the decentralisation of decision making. However, the decentralisation was not always in accordance with the requirements of shared leadership and shared responsibility for project success.

A fairly low variance, in the last column of Table 2, indicates a considerable consistency of teams (i.e. team agreement) in both courses. For the course MMSD, the variance ranges between 0.59 and 1.28. For the course SE, it ranges from 0.43 to 1.09 (with the exception of one team). These results are acceptable if one considers the aforementioned heterogeneity of some of development teams and different levels of prior knowledge. The teams with a lower team agreement score suffered from internal disagreements, which was also apparent in the quality of their solutions. The team agreement summary data for the individual dimensions show that the teams from both courses were most consistent when estimating three dimensions: learning, shared leadership and team orientation. Increased disparities can be observed in redundancy and autonomy.

A relatively low summary score and a low degree of team agreement in the redundancy dimension indicate that, within the teams, the students had different opinions about the extent to which the members can help and, if necessary, replace each other. Owing to differences in team members' knowledge and tight deadlines, it was often hard to find a replacement for a person in a team. A high variance in the autonomy dimension shows that the team members had different views on how other study obligations affect their work on the project.

The global team radar diagram in Figure 1 shows that there are no significant differences in dimension scores between both courses. Although Figure 1 actually shows three diagrams (the course SE, the course MMSD and the overall average), it gives an impression of a single diagram because of overlaps. It can be seen that the scores for autonomy and redundancy are much lower than the scores for other dimensions.

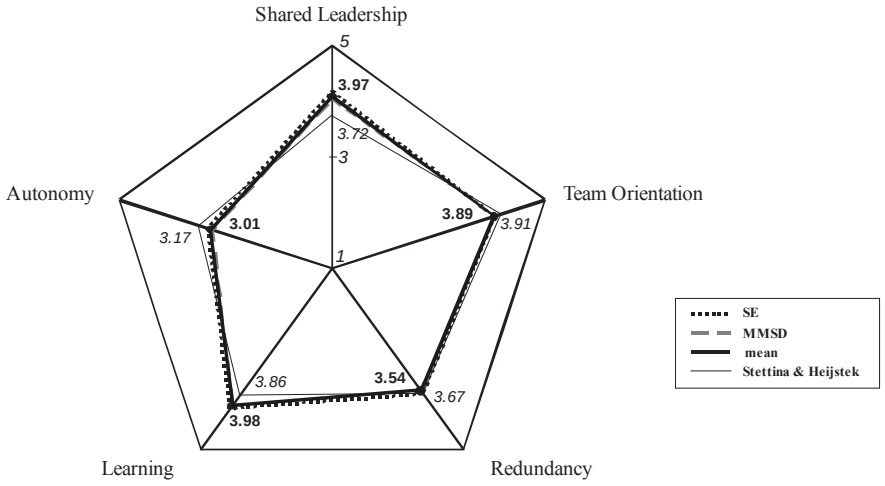

Figure 1: Global team radar.

The author compared the results with the results published by Stettina and Heijstek, which are also included in Figure 1 [15]. The comparison showed some similarities. The teams analysed by Stettina and Heijstek also gave the smallest score to autonomy. In both studies, the second worst score was assigned to redundancy. For the other dimensions, the order was not the same but there were no significant differences in score averages. Both in the SE group and the study of Stettina and Heijstek, the majority of teams assigned the highest score to learning.

The most significant difference between the two studies is the consistency of responses within individual teams. In the Stettina and Heijstek study, the variance for each team was between 0.06 and 0.33, indicating a very high degree of team agreement within teams. In the study here, the average team agreement was considerably smaller. However, this is understandable as the development teams analysed by Stettina and Heijstek consisted of experienced professionals who worked together significantly longer than did the student teams covered in the research for the present study.

CONCLUSION

In this article, the author has analysed the teamwork of the students who developed a relatively large capstone project within two courses using Scrum at the University of Ljubljana. The method developed by Moe et al was employed [14] and additionally adjusted by Stettina and Heijstek [15], which uses five dimensions of teamwork on the basis of survey data. Thirty-three student teams were surveyed and it was found that the students of both courses assigned the smallest score to autonomy, and the highest scores to learning and shared leadership. Most SE teams rated the learning dimension highest, while most MMSD teams assigned the highest score to shared leadership. The level of agreement within the teams was relatively high for the learning, shared leadership and team orientation dimensions. The development teams were less consistent in their assessment of the redundancy and autonomy dimensions.

The results were compared to those obtained by Stettina and Heijstek [15]. A correlation was found in the two dimensions with the smallest score and a considerable similarity in the average score of other dimensions. As expected, the professional developer teams surveyed by Stettina and Heijstek were more uniform in their assessments than the student teams in the study.

REFERENCES

1. West, D. and Grant, T., Agile Development: Mainstream Adoption Has Changed Agility, Forrester Research (2010), 22 February 2013, www.forrester.com/rb/Research/agile_development_mainstream_adoption_has_changed_agility/q/id/56100/t/2
2. Murphy, T., Duggan, J., Norton, D., Prentice, B., Plummer, D. and Landry, S., Predicts 2010: Agile and Cloud Impact Application Development Directions, Gartner (2009), 22 February 2013, www.gartner.com/DisplayDocument?id=1244514
3. Ambler, S.W., Has agile peaked? Let's look at the numbers. *Dr. Dobb's J.* (2008), 22 February 2013, www.ddj.com/architect/207600615?pgno=1
4. Layman, L., Williams, L., Slaten, K., Berenson, S. and Vouk, M., Addressing diverse needs through a balance of agile and plan-driven software development methodologies in the core software engineering course. *Inter. J. of Engng. Educ.*, 24, **4**, 659-670 (2008).

5. Rico, D.F. and Sayani, H.H., Use of agile methods in software engineering education. *Proc. Agile 2009 Conf.*, Chicago, USA, 174-179 (2009).
6. Schwaber, K., *Agile Project Management with Scrum*. Redmond: WA, Microsoft Press (2004).
7. VersionOne, State of Agile Survey 2011 (2011), 22 February 2013, www.versionone.com/pdf/2011_State_of_Agile_Development_Survey_Results.pdf
8. Mahnič, V., Teaching Scrum through team-project work: students' perceptions and teacher's observations. *Inter. J. of Engng. Educ.*, 26, **1**, 96-110 (2010).
9. Mahnič, V., A case study on agile estimating and planning using Scrum. *Electronics and Electrical Engng.*, 111, **5**, 123-128 (2011).
10. Mahnič, V. and Hovelja, T., On using planning poker for estimating user stories. *J. of Systems and Software*, 85, **9**, 2086-2095 (2012).
11. Mahnič, V., A capstone course on agile software development using Scrum. *IEEE Transactions on Educ.*, 55, **1**, 99-106 (2012).
12. Mahnič, V. and Rožanc, I., Students' perceptions of Scrum practices. *Proc. 35th Inter. Convention on Infor. and Communication Technol., Electronics and Microelectronics, MIPRO 2012*, Opatija, Croatia, 21-25 May 2012, 1364–1369 (2012).
13. Moe, N.B. and Dingsøyr, T., Scrum and team effectiveness: theory and practice. *Proc. 9th Inter. Conf. on Agile Processes in Software Engng. and Extreme Programming*, Limerick, Ireland, 11-20 (2008).
14. Moe, N.B., Dingsøyr, T. and Røyrvik, E.A., Putting agile teamwork to the test - a preliminary instrument for empirically assessing and improving agile software development. *Lecture Notes in Business Information Processing*, 31, 114-123 (2009).
15. Stettina, C.J. and Heijstek, W., Five agile factors: helping self-management to self-reflect. *Proc. 18th European System & Software Process Improvement and Innovation Conf. (EUROSPI 2011) (Communications in Computer and Information Science, 172, Springer)*, Roskilde, Denmark, 84-96 (2011).
16. Salas, E., Sims, D. and Burke, C., Is there a big five in teamwork? *Small GroupResearch*, 36, **5**, 555-599 (2005).
17. Hoegl, M. and Parboteeah, K.P., Autonomy and teamwork in innovative projects. *Human Resource Manag.*, 45, **1**, 67-79 (2006).