

Using Distributed Scrum for Supporting an Online Community - A Qualitative Descriptive Study of Students' Perceptions

Jörgen Söderback, Stefan Hrastinski, Lena-Maria Öberg

*Department of Computer and System Science, Mid Sweden University,
jorgen.soderback@miun.se, stefan.hrastinski@miun.se, lena-maria.oberg@miun.se*

Abstract

One purpose of higher education is to prepare students for a modern and ever-changing global society characterized by increasing complexity and collaborative environments. Scrum is an agile, widely used framework for project management dealing with the development of complex products. Scrum projects are conducted in small, empowered teams with intense communication, interaction and collaboration between the team members, facilitated by a servant-leader Scrum master. Scrum has been commonly used in professional software development and is also now being adopted in other areas, including education. There have been few studies of the application of Scrum in higher education and very few of them have studied distributed Scrum in an online context.

An online learning community has several positive effects for students such as increased learning, engagement, retention and lower risks for isolation and dropouts. Participating in and contributing to a team is dependent on a sense of community, which can be difficult to build up in a distributed environment where members are geographically dispersed and do not have the possibility to meet and communicate face to face.

This study examines to what extent and how distributed Scrum can support building an online learning community, from a student perspective. Twenty students, enrolled in an online course in distributed software development, participated in four Scrum projects as members of distributed Scrum teams, each team consisting of five students. Students' perceptions were investigated by conducting semi-structured interviews. The interview transcripts were analyzed according to Rovai's four dimensions of a classroom community. The results indicate that students were very satisfied with their distributed Scrum projects and that they experienced a high degree of flexibility during the projects. The Scrum process promoted and initiated communication and interaction among students and they learned how to communicate and collaborate effectively in an online environment. The transparency in Scrum was perceived as a key factor to open communication and effective collaboration and also contributed to increasing their motivation and engagement in the projects. Another interesting outcome of this study was understanding the importance of creating a team with members who are similar regarding competence level, ambition and preferences in working schedule.

Keywords

online education, Scrum, distributed, learning community, collaboration, transparency, interaction, motivation

Introduction

Higher education aims to prepare students for a modern, ever changing society and to give them the necessary tools to adapt to the changes (Ryan et al, 1996). The student of today needs skills, such as the ability to identify and solve problems, collaborate, communicate, interact and find and use information (Casner-Lotto & Barrington, 2006). The skills that students need to acquire are partly related to their future work role. In this paper, we focus on future software developers and, more specifically, students learning the concept of distributed software development in an online course. Typically, software developers work in projects and teams, and therefore it seems reasonable to encourage students to collaborate on project-based, real-world problems in real-world-similar contexts.

Participating in and contributing to a team is dependent on a sense of community. Members who report a strong sense of community are more likely to participate and help others in the team. Conversely, participating and helping others drive forward a sense of community (Wellman & Gulia, 1999). Collaboration has also been shown to decrease dropouts and increase retention (Rovai, 2002). Another aspect of collaboration in education is that students must learn how to work in teams (Snyder, 2009), although it is common that they have negative attitudes toward teamwork (Pope-Ruark, 2012). It is a challenge to design online courses that include learning about how to work in teams. In this paper, we investigate whether the use of a common software development framework, distributed Scrum, can be used to address this challenge in an online course in distributed software development.

Scrum is a framework for managing projects dealing with the development of complex products widely used in software development. The Scrum process is goal-driven, iterative and incremental, and cross-functional; self-organizing teams collaborate to reach their goals (Schwaber & Beedle, 2002). The software industry is also affected by the globalization which reveals that distributed development is an increasing trend (Shrivastava & Date, 2010). Communication, interaction and collaboration become a challenge when people are geographically spread out. Notably, distributed Scrum has been successfully used to support software development in global, online groups (Sutherland et al., 2007).

Drawing on the above, the aim of this paper is to examine if and how distributed Scrum can contribute towards the development of a sense of community in an online software development course. More specifically, we posit the following research question, from a student perspective: To what extent and how can distributed Scrum be used to support an online community? For the purpose of this study, we define community as “a group of people who are socially interdependent, who participate together in discussion and decision making, and who share certain practices that both define the community and are nurtured by it” (Bellah et al., 1985, p. 333). In the next section, we present the concepts of Scrum and community in more detail.

Theoretical background

This section presents the concept of community. Then, we introduce Scrum and provide an overview of how Scrum has been used and studied in a higher education context. Finally, we discuss the relationship between community and Scrum concepts.

Learning Community

Learning from a social constructivist perspective promotes the importance of building a learning community (Wenger, 1999). Rovai (2002) argues that a sense of community also can be built when working at a distance and that the medium is rarely the determining factor for building a sustainable community. Based on a literature review, Rovai (2002) suggests that a classroom community consists of four dimensions: spirit, trust, interaction and common expectations, all of which contribute to learning. The first dimension, spirit, describes social relations between members of the community such as friendship, commitment, enjoying each other's company and caring for each other. The second dimension, trust, is when members of the community trust and rely on each other. This dimension can be divided into two components: credibility, i.e. relying on what community members say, and benevolence, i.e. a concern for other members in the community. The third dimension, interaction, is categorized into task-driven or socio-emotional interaction. Task-related interaction is related to the specific problems or assignments that members of a community have to solve. Socio-emotional interaction is related to the interaction that builds and maintains social relations in order for the members of the community to relate to each other as human beings. The fourth dimension, common expectations, which in a learning community is learning, can be viewed as the member's self-motivation and commitment to learn and develop knowledge.

Scrum

Agile methods have evolved as a reaction to the traditional plan-driven software development methods and their difficulties in handling unstable requirements. According to Abbas et al. (2008), “[a]n Agile method is adaptive, iterative and incremental, and people-oriented” (p. 95). Agile methods promote open communication, transparency and visibility. A key factor to building agile teams is to establish the agile mindset within every team member (Stellman & Greene, 2014). Scrum is the most popular of the agile methods and was originally developed to address problems in the software development process, but it is also now being used in other areas

such as education, business, industry and the military (Damian et al., 2012; Rubin, 2012; Schwaber & Beedle, 2002). Scrum is based on an empirical process control model. Evidence comes from observations and control is achieved through transparency, inspection and adaptation (Schwaber & Beedle, 2002).

The Scrum process is divided into several iterations, called sprints. Every sprint has its own goal and the Scrum team focuses on the current sprint. The length of a sprint is between one and four weeks. Each sprint starts with a planning meeting which aims to set up a goal for the sprint to come (Schwaber & Beedle, 2002). Each day within a sprint normally starts with a short meeting up to 15 minutes: the daily Scrum. The purpose of daily Scrums is to make the progress visible and to build trust and spirit within the team. It also gives an opportunity to make necessary adjustments in the process. Every sprint ends with two meetings. The first meeting is called the sprint review meeting where the team demonstrates and delivers the part of the project that has been finished. The second meeting is called the retrospective meeting where the team inspect their work and make adjustments for continuous improvement. The Scrum process is described in detail and includes different roles, such as a product owner and “Scrum master”, and different ways of documenting.

A case study made by Mann and Maurer (2005) concluded an increase in customer satisfaction, more satisfied developers and less overtime after introduction of Scrum. Critical success factors are engaged and motivated team members and close cooperation with an engaged customer (Chow & Cao, 2008). Cho (2008) identifies issues and challenges with Scrum, those are however common also in other project management methods. Another limitation is that Scrum works best for small teams, three to nine developers, which makes it more suitable for smaller projects (Ionel, 2008). Most of the criticism of Scrum is related to how the process is being adapted, which could be interpreted as an indicator of the difficulty to adhere to Scrum.

Scrum in higher education

Previous research suggests that agile principles can benefit academic learning (e.g. Dewi & Muniandi, 2014; Kropp et al., 2014; Monett, 2013). Mahnic (2012) uses Scrum in a capstone (final year project) campus course on agile software development and states that students were overwhelmingly positive and confirms the previously reported benefits of Scrum. Scharff and Verma (2010) report that Scrum increases the chances of successful delivery in a mobile application development project. Ovesen (2013) argues, based on two case studies with students on campus, that the use of Scrum in problem-based learning projects in industrial design improves focus and team efficiency. Pope-Ruark (2012) suggests, after experimenting with Scrum in Web- software development projects on campus, that Scrum projects encourage trust, engagement and accountability among students.

Scharff et al. (2012) report from a project on the difficulties for students to adhere to Scrum in global software development (GSD) projects and suggest recommendations. Damian et al. (2012) describe initial challenges encountered when applying distributed Scrum practices on a globally distributed project. Paasivaara et al. (2013) conclude that Scrum supports collaboration and learning of important global software engineering skills, such as distributed communication, teamwork, building and maintaining trust, using appropriate tools and intercultural collaboration.

A Scrum team can be viewed as a small community. The four dimensions of a community (spirit, trust, interaction and common values (Rovai, 2002)) are somewhat similar to the values of a Scrum team: focus, courage, openness, commitment and respect (Schwaber & Beedle, 2002). Rovai (2002) has identified seven factors that are correlated to a sense of community: transactional distance, social presence, social equality, small group activities, group facilitation, teaching style and learning stage and community size. These factors correspond well to the implementation of distributed Scrum in an online education setting (see table 1). This suggests that distributed Scrum has the potential to contribute towards building and supporting a learning community.

Table 1. Factors that are correlated to a sense of community and Scrum

Community factor	Scrum
Transactional distance	The instructor as Scrum master and the learners as team members could enable a closer relationship between learners and instructors and, thus, decrease transactional distance.
Social presence	Social presence could be supported by intense communication and interaction encouraged in Scrum, entailing a feeling that others are there.

Social equality	Scrum describes the team as a unit in which all team members have the same role, indicating social equality.
Small group activities	Scrum teams are small and consist of three to nine team members, which are given tasks to meet deadlines.
Group facilitation	The Scrum master facilitates the team and eliminates obstacles related to both group tasks and team building.
Teaching style and learning stage	Scrum is based on transparency which gives the instructor valuable and continuous information about what students are doing in order to provide relevant and immediate feedback.
Community size	Scrum teams are small and consist of three to nine team members, making student-teacher ratio low, but still with opportunities for interaction between students.

Method

The Case

‘Distributed Software Development’ is a course in the fourth semester in a bachelor program called ‘Informatics with Focus on Systems Development’. The programme is held at Mid Sweden University. Twenty students took this five-week course during spring 2015. The course as well as the program is fully online. The purpose of the course is to give students an understanding of technical and social challenges related to distributed software development and knowledge on how to handle those challenges. The first week of the course students were introduced to various online collaboration tools and the Scrum framework. In the remaining four weeks they participated in a practical software development project.

Visual Studio Online (VSO) is a web-based tool for collaborative software development with support for several process templates. Students were introduced to VSO and were shown how to use the Scrum template to manage their projects. VSO can be connected with Visual Studio (VS) which is a software development tool with integrated support for Git, a distributed versioning control system. Git was used for versioning control and code sharing. The central repository was hosted within VSO. HipChat was introduced to handle group communication within the team. It is a group chat tool with support for notifications related to events in VSO, for example, when a new task is created, when a developer has made some code changes in the central repository or when a task is completed. The students were also introduced to various online collaborative tools for prototyping, modelling, mind mapping and documentation.

Students were allocated into four equally sized teams based on information gathered at the start of the course, where they estimated their own abilities and knowledge regarding, for example, computer programming, social ability, design skills. Four teachers were involved in the projects. One teacher had the role as a Product owner for all four projects. The other three teachers had the role of Scrum masters. One teacher was Scrum master for two teams and two teachers were Scrum masters for one team each. The Scrum projects were divided into three sprints, six working days each. Each sprint started with a sprint planning and ended with a sprint review and a sprint retrospective. Since time and resources were limited some meetings occurred the same day.

Each Developer team consisted of five students and they were responsible for organizing themselves. It was each team’s decision to choose which tools they were going to use. The first task was to create a contract and come to an agreement about communication, documentation, meetings, and tools. The students had to use Scrum to manage their projects, VSO using the Scrum template and Git for managing the work, code sharing and versioning management. Every team had daily Scrum meetings with their Scrum master where each team member was asked the following three questions: 1) What did I do yesterday that helped the team meet the sprint goal? 2) What will I do today to help the team meet the sprint goal? 3) Do I see any impediment that prevents me or the team from meeting the sprint goal?

Data Collection and Analysis

Purposeful sampling with maximum variation was used to gather as much information as possible (Creswell, 2002; Sandelowski, 2000). Variations in sex, age, work, time zone and family situation were considered during the sampling process. Semi-structured interviews were conducted with ten students within three weeks after the end of the course. The interviews lasted for 50-90 minutes and the purpose was to gather their perceptions of

being team members in distributed Scrum projects. The interviews were conducted via Skype and recorded using MP3 Skype Recorder. The recordings were then transcribed for further analysis.

Data was analyzed using qualitative content analysis by the first author. This is a commonly used method in qualitative descriptive research designs (Sandelowski, 2000). Qualitative content analysis is not only interested in the manifest data, but also the latent data, the underlying meaning of the text. In the first step of the analysis all transcriptions were read three times in order to make the reader more familiar with the transcripts. We used both axial and open coding to identify categories and themes underlying the categories. The data was initially categorized according to the four dimensions of community: spirit, trust, interaction and common expectations (Rovai, 2002). The empirical data was read and re-read iteratively, through open coding, in order to identify themes underlying each category (Strauss & Corbin 1997).

Context

The students were overwhelmingly positive to Scrum as a framework for project management. The course evaluation included many positive comments, such as "The Scrum project was inspiring and very, very enjoyable", "It has not felt like a school assignment", "It felt very realistic, like a real job" and "I almost cannot praise this course highly enough".

The students had worked in groups and projects earlier in the program, but in the previous projects had been responsible for forming their own teams. It seems that they had usually formed the same teams over and over again so they knew their own group members quite well, but had little knowledge about other students in the class. Even though the group had been studying together for almost four semesters they all reported that they had very little knowledge about the other members in their Scrum team. The team building processes started from scratch. Initially the students were frustrated about breaking up the old groups and were a little worried about working with new, relatively unknown, students. The major concern was about motivation and commitment and many students reported that they had bad experiences of classmates with low motivation and low commitment that did not contribute or became a part of the group resulting in an unsatisfactory team result and lower grades. After the initial reactions most were satisfied with the formation of the teams. They perceived the formation as contributing to the feeling of a real world project and that it helped them to develop their abilities in communication, interaction and collaboration. Students from group A however reported that they had a lot of problems with large differences in motivation, commitment, prior knowledge and different working schedules.

Results

We consider every Scrum team to be a small classroom community and the analysis regarding the four dimensions of a classroom community is based on the assumption that each team is considered to be a micro community in the course. As mentioned above, the students knew very few of their classmates before the Scrum project and the prior group formations were too small to be considered as learning communities, consisting only of two to three students.

Spirit

Students in two of the four teams reported that their team members were highly motivated. These participants were available for communication and interaction at almost any time of the day. They made it clear in advance if and when they would not be available. Every member of these two teams was committed and participated in most of the team's activities.

One member in Team C was having technical problems with the internet connection, which made him isolated from the rest of the team. "He was a member of the team but he wasn't actually a part of the team. He was more like an external resource, and that didn't feel good."

Team D had two members who had difficulties in communicating and interacting with the others because they were not able to work together simultaneously. One of them was located in another time-zone, nine hours behind the others, and she felt that she was not a member of the team. The other member was working during the day and was not available to the team until late afternoon. The daily Scrums helped team members to communicate and interact and they were very valuable for the members to bond and build friendship. There seems to be a connection between availability and team spirit: those who are available most of the time are

perceived as engaged, highly motivated and committed team members who seem to be able to establish a friendship with their classmates.

"We got to know each other quite well ... and we started to talk about more personal stuff and not just the Scrum project."

"There are always people that are more or less engaged and the most engaged members will have the largest impact on the project. But the team ... it felt ... yes, I would say that we were a team under the circumstances."

"It felt like every member didn't quite understand that it isn't the Scrum master's responsibility to assign tasks. Every team member has the responsibility to check out tasks and work with them until they are done. It feels like that was the hardest thing to understand for some members."

"You are obliged to be involved in a way and you have to engage. Maybe it's the same in other projects, but it's easier to sit in the background and not say anything than when you have to answer for what you have done and what you are planning to do in the daily Scrum. And that makes you ... you must commit yourself to ... "

Trust

In the beginning of the project students felt a little nervous about their own competence and were worrying about their contribution to the Scrum project. But when they understood that the team as a whole was responsible for the progress they felt more secure and began to trust in the team and its members.

"We had a downswing in the second sprint where we all felt a bit like: This is a real setback. But we were not afraid to express it to each other."

"I feel that there has been a trust within the group and that I could express a deviant opinion."

That made students feel comfortable to expose their weaknesses and to seek help among the other team members. The students were also concerned about their team members and were constantly offering them help when needed.

"You know what the other team members are doing so it's easy to ask if you can help someone with something."

The transparency in Scrum makes almost everything visible to team members. This can be used to see what other team members are working with which provides opportunities to ask for and offer help. Students say that trust in terms of both credibility and benevolence is high. The team is very important in Scrum and it seems to have a positive effect on students that they know that they always have backup from the team which makes it easier for them to seek for help. They are also very responsive to each other and are constantly offering help to other team members.

Interaction

The students felt that they had a lot of communication and interaction during the project. Team members are supposed to attend the daily Scrum meetings which establishes a foundation for communication and interaction. Every team held their daily Scrums via a Skype video conference. The daily Scrums are task-driven but the communication and interaction between team members were not limited to these meetings. After the daily Scrum, which is time-boxed to 15 minutes, a more informal meeting took place and it was a mixture of task-driven and socio-emotional interaction. Besides the planned meetings within Scrum, every team had meetings on demand several times a week. In three of the four teams the communication and interaction between team members went on all day. Every team member had a group chat open during the whole project and they had a lot of video calls.

"I think we were sitting together quite often ... daytime sharing a Skype call and screen sharing and were showing each other things and asking questions. It was actually a quite nice environment to work in. I would say that it was as close as you can get to working together in the same room"

"You have to be a bit social in Scrum. You have to participate, participation is ... that is what it takes ... You have to be engaged"

Common expectations: learning

The motivation was mostly high among the students but a few seemed to be less motivated which was reflected in the attendance, communication, interaction and their contribution to the team. Their fellow team members felt that these individuals were not a part of the team.

"You have an meeting at the same time every day where you say I've done this and it's good for the team to keep ... the team knows what to expect from you because if I take this, then I have to finish it and I will have to answer for what I've done or what I not have done."

"You take responsibility for what you do and you have to go back to the team and explain what you have done or why you haven't done anything. It's a good way to hold everyone responsible, so to speak. I think it's really good, really good"

Initial findings

The aim of this paper was to examine if and how distributed Scrum can contribute towards the development of a sense of community in an online software development course. This study indicates that distributed Scrum can be used to support the building of an online learning community. Daily meetings in Scrum initiate comprehensive communication and interaction between learners which is crucial for a learning community. Students perceive that the transparency in Scrum affects members in a way that makes the community stronger. It is important to put together members with similar motivation, competence and working schedule to create a strong sense of community. The instructor, in the role of Scrum master, must ensure that students fully understand the concept of Scrum and clarify the roles and responsibilities in Scrum to develop more self-directed learners.

Limitations and future research

This qualitative study is based upon semi-structured interviews with ten students and it is the students' perceptions and not actual observations which are the basis of the results. These results are based upon one course in distributed software development with twenty students in the fourth semester in a bachelor program. The overwhelmingly positive reactions from the students could be limited to this occasion, this context and the circumstances of the people involved. Thus, more research is needed that both tests and provides a more detailed understanding of the benefits and limitations of adopting agile methods, such as Scrum, in online higher education.

Since Scrum is a framework originally developed for software development it is easy to believe that the positive results in this study are valid only in this context, a university course in distributed software development. It would be interesting to see if these results are limited to courses in software development or if Scrum can support building online communities in other courses in different academic fields. The success of Scrum is dependent on highly motivated team members and it would be interesting to study to what extent a Scrum team can affect the team members' motivation due to the fact that members of a community can form the community and the community forms its members (Rovai, 2002). This study has viewed the Scrum team as a learning community and it would be interesting to investigate how these micro communities could be expanded to a larger community on a course level.

References

- Abbas, N., Gravell, A. M., & Wills, G. B. (2008). Historical roots of Agile methods: where did "Agile thinking" come from?. In *Agile Processes in Software Engineering and Extreme Programming* (pp. 94-103). Springer Berlin Heidelberg.
- Bellah, R. N., Madsen, R., Sullivan, W. M., Swidler, A., and Tipton, S. M. (1985). *Habits of the Heart: Individualism and commitment in American life*. New York: Harper and Row.
- Casner-Lotto, J., & Barrington, L. (2006). *Are They Really Ready to Work? Employers' Perspectives on the Basic Knowledge and Applied Skills of New Entrants to the 21st Century US Workforce*. Partnership for 21st Century Skills. 1 Massachusetts Avenue NW Suite 700, Washington, DC 20001.
- Proceedings of the 10th International Conference on Networked Learning 2016, Edited by: Cranmer S, Dohn NB, de Laat M, Ryberg T & Sime JA. ISBN 978-1-86220-324-2 403

- Cho, J. (2008). Issues and Challenges of agile software development with SCRUM. *Issues in Information Systems*, 9(2), 188-195.
- Chow, T., & Cao, D. B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6), 961-971.
- Damian, D., Lassenius, C., Paasivaara, M., Borici, A., Schröter, A. (2012) Teaching a globally project course using scrum practices. Published in: Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD), 9 June 2012 (pp. 30-34).
- Dewi, D. A., & Muniandy, M. (2014). The agility of agile methodology for teaching and learning activities. Paper presented at the Software Engineering Conference (MySEC), 2014 8th Malaysian.
- Ionel, N. (2008) Critical analysis of the Scrum Project Management methodology. *Annals of the University of Oradea, Economic Science Series*, 17(4), (pp.435–441).
- Kropp, M., Meier, A., Mateescu, M., & Zahn, C. (2014). Teaching and learning agile collaboration. Paper presented at the IEEE 27th Conference on Software Engineering Education and Training (CSEE&T).
- Mahnic, V., (2012). A capstone course on agile software development using scrum. *IEEE Transactions on Education*, Vol. 55, Issue: 1 (pp.99-106).
- Mann, C., Maurer, F.(2005), "A Case Study on the Impact of Scrum on Overtime and Customer Satisfaction", ADC, Agile Conference, Agile Conference 2005, pp. 70-79
- Monett, D.. Agile Project-Based Teaching and Learning. Paper presented at the Proceedings of the 11th International Conference on Software Engineering Research and Practice.
- Ovesen, N. (2013). Facilitating Problem-Based Learning in Teams with Scrum. In E. Bohemia, W. Ion, A. Kovacevic, J. Lawlor, M. McGrath, C. McMahon, B. Parkinson, G. Reilly, M. Ring, R. Simpson, ... D. Tormey (Eds.), *Design Education - Growing Our Future: Proceedings of the 15th International Conference on Engineering and Product Design Education*. (pp. 856). Chapter 861. Glasgow, UK: Design Society.
- Paasivaara, M., Lassenius, C., Damian, D., Rätty, P., Schröter, A., (2013). Teaching students global software engineering skills using distributed scrum
- Pope-Ruark, R., (2012). We scrum every day: Using scrum project management framework for group projects. *College teaching*, 60: (pp. 164-169).
- Rovai P. (2002) Building Sense of Community at a Distance. *The international review of research in open and distributed learning*. Vol. 3. No:1.
- Rubin, K. S. (2012). *Essential Scrum: A practical guide to the most popular Agile process*. Addison-Wesley.
- Ryan, G., Toohey, S., & Hughes, C. (1996). The purpose, value and structure of the practicum in higher education: A literature review. *Higher Education*, 31(3), 355-377.
- Scharff, C. Heng, S., Kulkarni, V. (2012). On the difficulties for students to adhere to scrum on global software development projects: Preliminary results. Published in Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD), 2012, (pp.25-29).
- Scharff, C., Verma R.(2010) Scrum to support mobile application development projects in a just-in-time learning context. In proceeding CHASE '10 Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering. (pp. 25-31)
- Schwaber, K., Beedle, M. (2002). *Agile Software Development with Scrum*. Pearson International Edition.
- Shrivastava, S.V., Date, H., (2010) Distributed agile software development: A review. *Journal of Computer Science and Engineering*, Volume 1, Issue 1, (pp. 10-17).
- Snyder, L. G. (2009). Teaching Teams about Teamwork: Preparation, Practice, and Performance Review. *Business Communication Quarterly*, 72(1), (pp. 74-79)
- Stellman, A., & Greene, J. (2005). *Applied software project management*. "O'Reilly Media, Inc."
- Strauss, A. L. & Corbin, J. M. (Eds.). (1997). *Grounded theory in practice*. Sage.
- Sutherland, J., Viktorov, A., Blount, J., & Puntikov, N. (2007). Distributed scrum: Agile project management with outsourced development teams. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on* (pp. 274a-274a). IEEE.
- Wellman, B., & Gulia, M. (1999). Virtual communities as communities: Net surfers don't ride alone. In M. Smith & P. Kollock (Eds.), *Communities in cyberspace* (pp. 167-194). London, England: Routledge.
- Wenger E., (1999). *Communities of Practice - Learning, Meaning and Identity*. Cambridge University Press