Journal of Information Systems Education

Volume 29
Issue 2
Spring 2018

Agile in Teaching and Learning: Conceptual Framework and Research Agenda

Jason H. Sharp and Guido Lang

Recommended Citation: Sharp, J. H. & Lang, G. (2018). Agile in Teaching and Learning: Conceptual Framework and Research Agenda. *Journal of Information Systems Education,* 29(2), 45-52.

Article Link: http://jise.org/Volume29/n2/JISEv29n2p45.html

Initial Submission: 1 April 2018
Accepted: 6 April 2018
Published: 13 June 2018

ISSN: 2574-3872 (Online) 1055-3096 (Print)

Agile in Teaching and Learning: Conceptual Framework and Research Agenda

Jason H. Sharp
Department of Marketing and Computer Information Systems
Tarleton State University
Stephenville, TX 76402, USA
jsharp@tarleton.edu

Guido Lang
Department of Computer Information Systems
Quinnipiac University
Hamden, CT 06518, USA
guido.lang@quinnipiac.edu

ABSTRACT

Agile software development methods are widespread in industry, and there is a wealth of academic research and practitioner publications currently available from this perspective. With the rise of Agile within companies worldwide, it is increasingly important for information systems education to keep up with this trend to ensure curriculum and courses are up-to-date. Students in the computing disciplines must be prepared to enter a job market where Agile is commonplace. As such, the topic of Agile in teaching and learning is critically important. The current special issue includes a rich collection of articles providing information systems educators with research-based, practical approaches for both teaching Agile ("the what") and using Agile as a pedagogical approach ("the how"). In an effort to assist information systems educators categorize the growing amount of literature related to Agile in teaching and learning, a conceptual framework is provided which places the literature along the two axes of pedagogy ("the how") and the content ("the what") ranging from other, non-Agile to Agile. Finally, the authors present a call for future research integrating Agile on a meta-level in the course development process. We hope that this special issue inspires educators and researchers to consider integrating Agile into their teaching and learning.

Keywords: Agile, Scrum, Pedagogy, Framework, Pair programming, Self-regulated learning, Cooperative learning

1. INTRODUCTION

In recent years, the use of Agile software development methods, or Agile for short, has become increasingly popular as a way of producing software in a lighter, quicker, more people-centered manner. Agile represents an emerging set of software development methodologies based upon the concepts of adaptability and flexibility (Abrahamson et al., 2003). Since the release of the Agile Manifesto in 2001, the popularity and use of Agile has continued to grow. Specific Agile methodologies such as Scrum and eXtreme Programming have gained widespread recognition. As such, there has been a wealth of academic research published related to the implementation of Agile in industry.

With the end roads that Agile has established in industry and subsequent industry-related academic research, its impact on information systems education and the preparation of graduates in the computing disciplines is growing in importance. The focus of this special issue is the implementation of Agile in the classroom as it relates to both teaching and learning. Articles included in the special issue describe the implementation of Agile into the classroom, how Agile principles and practices improve teaching and learning, and what the future of Agile in information systems education potentially looks like.

2. CURRENT RESEARCH ISSUES IN AGILE IN TEACHING AND LEARNING

The concept of using Agile as an approach for teaching and learning is not novel (e.g., Andersson and Bendix, 2005, 2006; Chun, 2004; Lang, 2017; Razmov and Anderson, 2006; Vuokko and Berg, 2007). However, while related articles appear periodically in such publications as the *Journal of Information Systems Education (JISE)* and other information systems education publication outlets, to the knowledge of the special issue co-editors, no attempt to compile research on Agile in teaching and learning into a single journal issue currently exists.

A search of JISE articles reveals that the first article proposing an Agile teaching approach appeared in McBride (2005) who applied the values of eXtreme programming (i.e., good communication, simplicity, feedback, and courage) as a model to teach an e-commerce course. In that same year, McAvoy and Sammon (2005) developed an adoption assessment matrix to assist in the selection of the appropriate Agile method for use in specific software projects. A component of the study proposed a pedagogical approach based on active learning to "improve the student's knowledge of the adoption of Agile methods" (p. 409) through participation in critical adoption factors workshops.

An increasingly common practice is to introduce Agile into existing courses alongside traditional approaches. Due to the growing popularity of Agile project management approaches like Scrum, the implementation of Agile is a natural fit for project management and capstone courses, which are common in computing curricula (e.g., Laplante, 2006; Morien, 2004; Ramakrishnan, 2009). Schwalbe (2012) provides a brief, yet thorough, approach to incorporating Scrum to manage a project based upon the process groups of the Project Management Institute's Project Management Body of Knowledge guide. Building on the work of Schwalbe (2012), Landry and McDaniel (2016) put theory into practice by developing course content, assignments, and assessments for implementing Agile into a traditional project management course. Similarly, Baird and Riggins (2012) implement a hybrid, project management methodology consisting of traditional project management (waterfall) and Agile project management principles (Scrum) in a Computer Information Systems (CIS) capstone course in an effort to capture students' satisfaction and perceptions of such a hybrid project management approach.

Much like the limited exposure to Agile in other information systems courses, May, York, and Lending (2016) argue that most systems analysis and design textbooks provide only cursory content on Agile, in particular Scrum. In an attempt to provide students with a fuller experience of the Scrum framework and its element, they implement the "Ball Game" into their systems analysis and design course. As the authors note, "the primary purpose of this exercise is for students to experience for themselves the effects of a self-organizing team" in an effort to "drive home the various elements of the Scrum framework and how it differs from traditional approaches" (p. 89). Also citing the theoretical nature of Agile interspersed within information systems curricula, Weber (2016) proposes the pairing of the systems analysis and design course with a web-mobile programming course to allow students to apply the concepts of Agile, not simply to read about it – that is, to provide the students with a "real-world" experience. As such, the use of Performance Learning (Podeschi, 2015) affords a means to "provide a more accurate representation and direct opportunities to practice concepts learned in the classroom" (p. 4). The goal is that "course content is immediately applied by students utilizing new acquired skills while working on real-world projects for real-world third-party stakeholders with real-world risk and rewards" (p. 4).

The brief review of previous research on Agile in teaching and learning in JISE reveals that while some studies report on teaching about Agile software development (e.g., McAvoy and Sammon, 2005), others integrate pedagogical interventions based on the principles of Agile software development without necessarily teaching about Agile itself (e.g., McBride, 2005). The differentiation here is between what is being taught (i.e., the content) and how it is being taught (i.e., the pedagogy). Since both the content and the pedagogy can include varying degrees of Agile practices, one can place previous and future studies along the two axes of pedagogy (from other, non-Agile to Agile) and content (from other, non-Agile to Agile). Figure 1 depicts the resulting conceptual framework.

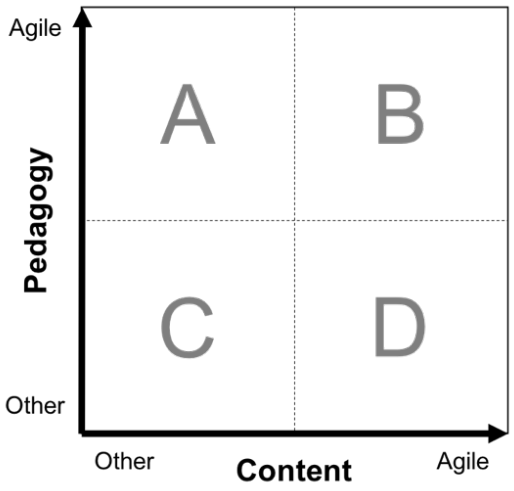

Figure 1. Conceptual Framework for Integrating Agile in Teaching and Learning

The content and pedagogy of a curriculum, course, lesson, or exercise may utilize varying degrees of Agile principles. For example, a course on Agile software development that is taught in a traditional lecture format would be placed in the lower right quadrant ("D") in the conceptual framework. On the other hand, the same course on Agile software development that is taught by having students engage in Agile techniques, such as Scrum, would be placed in the upper right quadrant ("B"). Consequently a course on cybersecurity that is taught using cases would be placed in the lower left quadrant ("C"), while the same course that is taught by having students engage in Agile techniques, such as pair programming, would be placed in the upper left corner ("A"). Naturally, the borders between Agile and other pedagogy as well as between Agile and other content are fluid. Based on the learning needs of the students and the preferences of the instructor, certain aspects of Agile pedagogy may be used in an otherwise traditional class. For example, an instructor may use iterative development approaches or reflection journals without fully committing to a full Agile pedagogy. Likewise, the concepts of Agile software development may be covered only in parts of a course or a lesson. The conceptual framework thus aims to encourage educators and researchers in information systems and beyond to integrate Agile development into the pedagogy and content of their courses.

3. INTRODUCTION TO SPECIAL ISSUE ARTICLES

This special issue, entitled, "Agile in Teaching and Learning," contains six articles, all of which utilize various Agile pedagogical interventions. The first three articles use Agile pedagogy to teach content that is not about Agile methodology itself. The last three articles use Agile pedagogy to teach students about Agile methodology. Figure 2 places the studies in the corresponding quadrants of the conceptual framework.

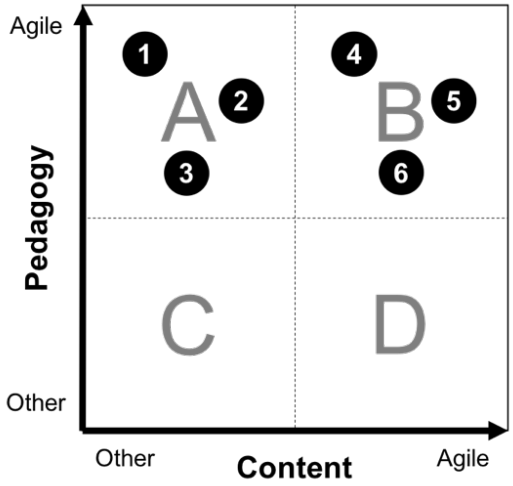

Figure 2. Conceptual Framework with Studies from this Special Issue

3.1 Measuring Agile Attitudes with Pair Programming

With the introduction of eXtreme Programming in the late 1990s (Beck, 1999), the implementation of pair programming to enhance student learning, increase student confidence, and improve student attitude has been the subject of numerous articles in the computing disciplines (e.g., Williams and Kessler, 2000, 2001; Williams et al., 2002). The first article in the special issue, "Do Pair Programming Approaches Transcend Code? Measuring Agile Attitudes in Diverse Information Systems Courses" by Kuanchin Chen and Alan Rea, continues this rich tradition of study. The stated objectives of the paper include: (1) to examine how participant attitudes and perceived benefits of pair programming are related to quality of solution and (2) to study if the quality of solution through pair programming varies across multiple disciplines. Additionally, the authors are interested in examining how pair programming affects areas of IS study beyond software development. The authors study 74 student participants across 4 IS classes (i.e., introduction to computing, programming, analytics, and data mining) including freshmen to seniors, and both IS majors and non-majors. The results indicate interesting findings related to attitude differences between genders when working solo and in pairs; quality of solution based upon age, motivation, working solo or in pairs, and perceived sense of accomplishment; and impact of experimental setting between the freshman-level course (introduction to computing) and the senior-level course (data mining). The authors suggest that pair programming may not necessarily be a "key driver to affect attitude changes" or "consistently associated with improvement of solution quality." Furthermore, the type of "subject matter is less important to solution quality improvement compared to whether one is involved in pair programming" and "perceptions on outcomes vary across different level of academic preparation." In conclusion, the authors assert that the study supports the idea that pair programming is implementable in courses beyond software development and lends a new perspective on gender differences, and other attitude measures beyond confidence may affect solution quality improvement more than previously thought.

3.2 Fostering Self-Regulated Learning through Scrum-Based Environment

The study of self-regulation on learning is certainly not novel (e.g., Zimmerman, 1986). The overall argument is that self-regulated learners possess a greater degree of understanding of how to use various strategies to enhance learning, greater metacognitive skill, and greater motivational beliefs and attitudes (Klassen, Krawchuk, and Rajani, 2008; Wolters, 2003). Recent studies examine self-regulation as it relates to procrastination (e.g., Dunn, 2013; Steel and Klingsieck, 2016; Waschle et al., 2014; Wolters, 2003), online instruction (Sharp and Sharp, 2016), and programming (Hui and Umar, 2011; Pedrosa et al., 2016). The second article, "Scrum-Based Learning Environment: Fostering Self-Regulated Learning" by Tanya Linden, details how Scrum is used to facilitate self-regulated learning in an introductory programming course within the Doubtfire learning management system (LMS). The author presents an interesting, non-traditional approach in contrast to the traditional approach to teaching programming. The study seeks to answer the following questions: (1) what is students' acceptance of our non-traditional approach using Scrum to facilitate the acquisition of self-regulated learning skills? and (2) does our non-traditional Scrum based approach improve students' pass rates in the introductory programming subject? In particular, the study focuses on perceived autonomy and perceived competence. According to the author, the measurement of perceived autonomy demonstrates "that the majority of our students are in favor of the environment that allows them to work using Scrum approach and supports self-regulated learning" thus positively answering the first research question. In terms of perceived competence, an evaluation of failure rates over the past three semesters reveals that the non-traditional approach did not improve pass rates. In conclusion, the author states, "although this non-traditional approach benefited self-regulated learners, it did not improve motivation of disinterested students and did not have any positive effect on the ratio of pass/failure rates."

3.3 Fostering Cooperative Learning with Scrum

In line with previous research examining the use of Collaborative Learning and Agile (e.g., Maguire et al., 2014), the third article, "Fostering Cooperative Learning with Scrum in a Semi-Capstone Systems Analysis and Design Course" by Alejandra J. Magana, Ying Ying Seah, and Paul Thomas, addresses the integration of Scrum principles and Cooperative Learning guidelines into a systems analysis and design course. In doing so, the authors hope to better facilitate teamwork, communication, and problem-solving while at the same time teaching the appropriate methods of systems analysis and design. This study seeks to answer the following research questions: (1) what are the students' levels of achievement in a system analysis and design course that integrates learning and

Agile methods through a semester-long project? (2) what are students' reflections on their learning and performance as a team working on a semester-long project facilitated with Agile methods? and (3) what are students' perceptions of a systems analysis and design course that integrates Cooperative Learning and Agile methods through a semester-long project? Using Cooperative Learning as a pedagogical framework, the authors implemented the study with 2 cohorts of 100 students each, divided into 5-person teams, consisting primarily of undergraduate computer and information technology majors. The first cohort followed an overlapped approach, while the second cohort followed a delayed approach. To address the research questions, the authors examine three constructs: (1) student performance in the course, (2) student reflections on their team performance, and (3) student overall perceptions of the teaching approach. The results indicate statistically significant differences on student performance in the course between the use of the overlapped approach and the delayed approach, the identification of five themes related to student reflection on team performance, and that student overall perceptions of the teaching approach were mixed between cohorts. The authors conclude that "Cooperative Learning combined with Scrum can effectively guide students in analyzing and designing software solutions."

3.4 A Three Cohort Study of Role-Play Instruction for Agile Project Management

Role-play exercises have a rich history in information systems education (e.g., Freeman, 2003; Mitri and Cole, 2007; Shen, Nicholson, and Nicholson, 2015) where they have often been found to be superior to traditional methods of instruction (Kerr, Troth, and Pickering, 2003). In line with this research, the fourth article, "A Three Cohort Study of Role-Play Instruction for Agile Project Management" by Kurt Schmitz, introduces and evaluates the efficacy of a role-play exercise called "Scrummy" which aims to help students better understand Agile project management through experiential learning. The role-play exercise was designed to be completed alternatively in a single 2.5-hour class, two 75-minute classes, or three 50-minute classes. The role-play exercise adapts the Scrum Software Development process in an abbreviated form, with students working in teams of 4 to 6 to complete up to 4 sprints (of about 30 minutes each). Throughout the sprints, students are faced with real-world challenges such as requirement changes and scope creep. Although the role-play exercise was designed for undergraduate healthcare informatics students, the sample project can be easily adapted for different learning contexts. The author evaluated the efficacy of the role-play exercise in three undergraduate classes using a combination of pre- and post-test self-efficacy measures as well as content-matched exam questions. Compared to a traditional lecture method, the role-play exercise was found to produce significantly higher levels of students' self-efficacy and actual comprehension of Agile concepts. As a result, Schmitz concludes that this study "demonstrates the comparative efficacy of role-play over traditional reading and lecture for the concepts of Agile project management."

3.5 Origami: An Active Learning Exercise from Scrum Project Management

Active learning "requires students to do meaningful learning activities and think about what they are doing" (Prince, 2014, p. 223). As such, active learning is well-suited to teach Agile methodologies such as Scrum. Although several active learning exercises have been proposed to teach Scrum, many of them do not introduce requirement changes (e.g., Fernandes and Sousa, 2010; Paasivaara et al., 2014; Von Wangenheim, Savi, and Borgatto, 2013). Given that requirement changes form a core tenet of the Agile principles ("Responding to change over following a plan"), more active learning exercises that allow students to experience a change in requirements are needed. To address this need, the fifth article, "Origami: An Active Learning Exercise for Scrum Project Management" by Christopher Sibona, Saba Pourreza, and Stephen Hill, presents and evaluates a Scrum exercise that allows students to experience a change in requirements that varies from the initial plan due to their progress in the task completion. The active learning exercise can be completed in a 50-minute class. The exercise asks students to create various origami over the course of three simulated days (a day lasts five minutes in the exercise), including sprint planning, daily scrum meetings, sprint reviews, and a sprint retrospective. To compare the effectiveness of the Scrum exercise to a traditional lecture on Scrum, the authors randomly assigned five classes to receive either the lecture first (followed by the activity) or the activity first (followed by the lecture). Based on students' self-reported knowledge and perceived engagement, the authors conclude that "the lecture followed by activity is the preferred approach when two days are allowed. If time permits one day for this lesson then the activity is preferred as students found the activity to be more engaging."

3.6 Coping with Uncertainty in an Agile Systems Development Course

Uncertainty refers to the emotion caused by ambiguity, which in turn can be caused by some combination of cues in ambiguous situations: (1) situations that do not present any known cues are considered new, (2) situations that have a great number of cues are considered complex, and (3) situations that exhibit conflicting cues are considered contradictory (Budner, 1962). In the context of information systems development, uncertainty has been found to negatively affect process performance and product quality (Jun, Qiuzhen, and Qingguo, 2011). Agile systems development arose partially out of the need to cope with uncertainty in terms of changing technical, organizational, and business environments in systems development projects. While traditional teaching methods strive to reduce uncertainty for students, the sixth article, "Coping with Uncertainty in an Agile Systems Development Course" by Toni Taipalus, Ville Seppänen, and Maritta Pirhonen, presents evidence for the pedagogical benefits of creating uncertainty. As part of an undergraduate systems development course for computer science and information systems students, the authors asked students to self-select into Scrum teams and complete a realistic, semester-long systems development project. The authors consciously integrated causes for ambiguity in the course, including minimum amounts of teacher interaction, an ambiguous target domain introduction, a large and complex project, and changes in organizational,

business, and technical environments. At the end of the course, students completed a survey measuring perceived uncertainty and process performance. Product quality was measured using the final grade given for the project deliverables. Factor analyses indicated that students developed three distinct coping strategies with varying success: (a) versatile performers stepped out of their Scrum roles to 'do what needs to be done,' i.e., to contribute outside of their Scrum role and outside of their area of expertise; (b) determined performers stepped out of their Scrum roles to 'do what they know,' i.e., to contribute outside of their Scrum role but within their area of expertise; while (c) obedient performers stayed within their prescribed Scrum roles to 'do what they've been told.' Findings indicate that obedient performers significantly fared better in terms of process performance and product quality. In other words, "students who rigorously followed the Scrum guidelines and practices were better equipped to deal with the changes in requirements and other sources of uncertainty." As a result, instructors should stress the importance of narrowly following the Scrum methodology, especially if students are inexperienced in applying Scrum.

4. CHALLENGE TO READERS

It appears that Agile is not going away any time soon, and with the continued rise of popularity in industry, the challenge to information systems educators is to ensure that students are well prepared to enter the workforce where Agile is becoming predominant. We strongly believe in the potential of the Agile methodology to improve teaching and learning. While previous research, including the articles presented in this special issue, combined various aspects of Agile pedagogy and Agile content, to the best of the authors' knowledge, no systematic research has been conducted about the use of Agile for course development itself. Figure 3 depicts the integrated conceptual framework for Agile in teaching and learning.

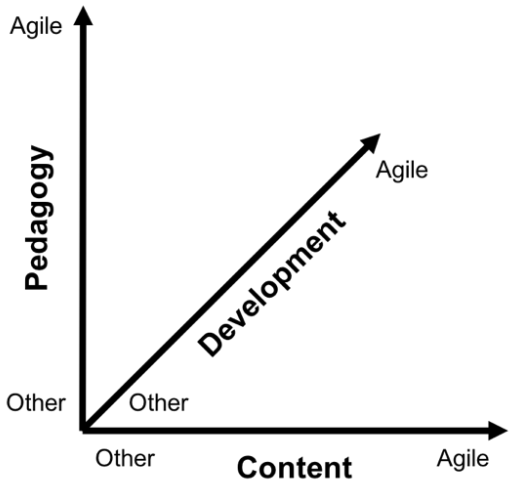

Figure 3. Integrated Conceptual Framework for Agile in Teaching and Learning

For example, a course on Agile systems development that is taught by having students engage in Agile techniques (previously quadrant "B") may be developed in two-week sprints with regular feedback from students being integrated to improve the learning experience. In essence, the third dimension reflects the application of Agile principles on a meta-level. Given that instructors face large amounts of uncertainty regarding the needs and capabilities of the students prior to or at the beginning of a course, it appears that Agile principles may be useful in the course development process.

We hope that this special issue inspires educators and researchers to consider integrating Agile into their teaching and learning. While the articles in this special issue present novel contributions to our understanding of Agile in teaching and learning, it is clear that further research is needed to fully understand and apply Agile software development techniques in the context of information systems education.

5. REFERENCES

Abrahamson, P., Warsta, J., Sippon, S. T., & Ronkainen, J. (2003). New Directions on Agile Methods: A Comparative Analysis. *Proceedings of the 25th International Conference on Software Engineering*, 244-254.

Andersson, R. & Bendix, L. (2005). eXtreme Teaching. *Proceedings of 3:e Pedagogiska Inspirationskonferensen*, LTH, Lund Institute of Technology.

Andersson, R. & Bendix, L. (2006). Towards a Set of eXtreme Teaching Practices. In Salakoski, T., T. Mantyla, & M. Lasko (eds.) *Proceedings of Koli Calling 2005. 5th Koli Calling Conference on Computer Science Education*, November 17-20, 2005, Koli, Finland. TUCS Genera Publication 41, January 2006, 33-40.

Baird, A. & Riggins, F. J. (2012). Planning and Sprinting: Use of a Hybrid Project Management Methodology within a CIS Capstone Course. *Journal of Information Systems Education*, 23(3), 243-257.

Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Boston: Addison-Wesley Longman.

Budner, S. (1962). Intolerance of Ambiguity as a Personality Variable. *Journal of Personality*, 30, 29-50.

Chun, A. H. W. (2004). The Agile Teaching/Learning Methodology and its E-Learning Platform. *Lecture Notes in Computer Science*, 3143, Springer-Verlag Heidelberg, 11-18.

Dunn, K. (2014). Why Wait? The Influence of Academic Self-Regulation, Intrinsic Motivation, and Statistics Anxiety on Procrastination in Online Statistics. *Innovative Higher Education*, 39, 33-44.

Fernandes, J. M. & Sousa, S. M. (2010). Playscrum – A Card Game to Learn the Scrum Agile Method. *Games and Virtual Worlds for Serious Applications (VS-GAMES), 2010 Second International Conference*, 52-59.

Freeman, L. A. (2003). Simulation and Role Playing with LEGO Blocks. *Journal of Information Systems Education*, 14(2), 137-144.

Hui, T. H. & Umar, I. N. (2011). Does a Combination of Metaphor and Pairing Activity Help Programming Performance of Students with Different Self-Regulated Learning Level? *The Turkish Online Journal of Educational Technology*, 19(4), 121-129.

Jun, L., Qiuzhen, W., & Qingguo, M. (2011). The Effects of Project Uncertainty and Risk Management on IS Development Project Performance: A Vendor Perspective. *International Journal of Project Management, 29*(7), 923-933.

Kerr, D., Troth, A., & Pickering, A. (2003). The Use of Role-Playing to Help Students Understand Information Systems Case Studies. *Journal of Information Systems Education*, 14(2), 167-172.

Klassen, R. M., Krawchuk, L. L., & Rajani, S. (2008). Academic Procrastination of Undergraduates: Low Self-Efficacy to Self-Regulate Predicts Higher Levels of Procrastination. *Contemporary Educational Psychology*, 33, 915-931.

Landry, J. P. & McDaniel, R. (2016) Agile Preparation within a Traditional Project Management Course. *Information Systems Education Journal*, 14(6), 27-33.

Lang, G. (2017). Agile Learning: Sprinting through the Semester. *Information Systems Education Journal*, 15(3), 14-21.

Laplante, P. A. (2006). An Agile, Graduate, Software Studio Course. *IEEE Transactions on Education*, 49(4), 417-419.

Maguire, P., Maguire, R., Hyland, P., & Patrick, M. (2014). Enhancing Collaborative Learning Using Pair Programming: Who Benefits? *All Ireland Journal of Teaching and Learning in Higher Education*, 6(2), 14111-14124.

May, J., York, J., & Lending, D. (2016). Play Ball: Bringing Scrum into the Classroom. *Journal of Information Systems Education*, 27(2), 87-92.

McAvoy, J. & Sammon, D. (2005). Agile Methodology Adoption Decisions: An Innovative Approach to Teaching and Learning. *Journal of Information Systems Education*, 16(4), 409-420.

McBride, N. K. (2005). A Student-Driven Approach to Teaching E-Commerce. *Journal of Information Systems Education*, 16(1), 75-83.

Mitri, M. & Cole, C. (2007). A Systems Analysis Role Play Case: We Sell Stuff, Inc. *Journal of Information Systems Education*, 18(2), 163-168.

Morien, R. I. (2004). Insights into Using Agile Development Methods in Student Final Year Projects. *Issues in Informing Science and Information Technology*, 605-624.

Paasivaara, M., Heikkilä, V., Lassenius, C., & Toivola, T. (2014). Teaching Students Scrum Using Lego Blocks. *Companion Proceedings of the 36th International Conference on Software Engineering*, 382-391.

Pedrosa, D., Cravino, J., Morgado, L., & Barreira, C. (2016). Self-Regulated Learning in Computer Programming: Strategies Students Adopted During an Assignment. *Proceedings of the International Conference on Immersive Learning*, 87-101.

Podeschi, R. (2016). Building I.S. Professionals through a Real-World Client Project in a Database Application Development Course. *Information Systems Education Journal*, 14(6), 34-30.

Prince, M. (2004). Does Active Learning Work? A Review of the Research. *Journal of Engineering Education*, 93(3), 223-231.

Ramakrishnan, S. (2009). Innovation and Scaling up Agile Software Engineering Projects. *Issues in Informing Science and Information Technology*, 6, 557-575.

Razmov, V. & Anderson, R. (2006). Experiences with Agile Teaching in Project-Based Courses. *Proceedings of the ASEE Annual Conference and Exposition.*

Schwalbe, K. (2012). Managing a Project Using an Agile Approach and the PMBOK® Guide. *Proceedings of the Information Systems Educators Conference*, 29(1985), 1-8.

Shen, Y., Nicholson, J., & Nicholson, D. (2015). Using a Group Role-Play Exercise to Engage Students in Learning Business Processes and ERP. *Journal of Information Systems Education*, 26(4), 265-280.

Sharp, L. A. & Sharp, J. H. (2016). Enhancing Student Success in Online Learning Experiences through Use of Self-Regulation. *Journal on Excellence in College Teaching*, 27(2), 57-75.

Steel, P. & Klingsieck, K. B. (2016). Academic Procrastination: Psychological Antecedents Revisited. *Australian Psychologist*, 51, 36-46.

Von Wangenheim, C. G., Savi, R., & Borgatto, A. F. (2013). Scrumia – An Educational Game for Teaching Scrum in Computing Courses. *Journal of Systems and Software*, 86(10), 2675-2687.

Vuokko, R. & Berg, P. (2007). Experimenting with eXtreme Teaching Method – Assessing Students' and Teachers' Experiences. *Issues in Informing Science and Information Technology*, 4, 523-534.

Waschle, K., Allgaier, A., Lachner, A., Fink, S., & Nuckles, M. (2014). Procrastination and Self-Efficacy: Tracing Vicious and Virtuous Circles in Self-Regulated Learning. *Learning and Instruction*, 29, 103-114.

Weber, E. V. (2016). Performance Learning of Agile Methodology Using Paired Courses of Systems Analysis and Design and Web/Mobile Programming. *Proceedings of the EDSIG Conference on Information Systems and Computing Education*, 2(4034), 1-8.

Williams, L. A. & Kessler, R. R. (2000). All I Ever Needed to Know about Pair Programming I Learned in Kindergarten. *Communications of the ACM*, 43(5), 108-114.

Williams, L. A. & Kessler, R. R. (2001). Experiments with Industry's "Pair-Programming" Model in the Computer Science Classroom. *Computer Science Education*, 11(1), 7-20.

Williams, L. A., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In Support of Pair Programming in the Introductory Computer Science Course. *Computer Science Education*, 12(3), 197-212.

Wolters, C. A. (2003). Understanding Procrastination from a Self-Regulated Learning Perspective. *Journal of Educational Psychology*, 95(1), 179-187.

Zimmerman, B. J. (1986). Becoming a Self-Regulated Learner: Which are the Key Subprocesses?. *Contemporary Educational Psychology*, 11(4), 307-313.

AUTHOR BIOGRAPHIES

Jason H. Sharp is an Associate Professor of Computer Information Systems at Tarleton State University. He earned his Ph.D. from the University of North Texas. Prior to entering academia, he worked as a systems support specialist and developed custom database solutions for small business. His research focuses on systems and software development methods and instructional design and technology. His work has appeared in such journals as *Information Systems Education Journal*, *Journal of Information Technology Education: Innovations in Practice*, and *The DATA BASE for Advances in Information Systems*, as well as in the proceedings of numerous international, national, and regional conferences.

Guido Lang is an Associate Professor of Computer Information Systems at Quinnipiac University. He earned his Ph.D. with a specialization in information systems from The City University of New York. He is an academic entrepreneur and has founded successful startups in the technology, publishing, and healthcare sectors. His research focuses on curriculum development and pedagogy in information systems and computing education. His work has won awards from the Education Special Interest Group of the Association for Information Technology Professionals and appeared in such journals as *Decision Support Systems*, *Information Systems Education Journal*, *Journal of Research on Technology in Education*, and *Journal of Computer Information Systems*.

Information Systems & Computing
Academic Professionals

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 2574-3872