Journal of Information Systems Education

Volume 29
Issue 2
Spring 2018

Origami: An Active Learning Exercise for Scrum Project Management

Christopher Sibona, Saba Pourreza, and Stephen Hill

Recommended Citation: Sibona, C., Pourreza, S., & Hill, S. (2018). Origami: An Active Learning Exercise for Scrum Project Management. *Journal of Information Systems Education,* 29(2), pp. 105-116.

Article Link: http://jise.org/Volume29/n2/JISEv29n2p105.html

Initial Submission: 16 August 2017
Accepted: 30 January 2018
Abstract Posted Online: 21 March 2018
Published: 13 June 2018

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

Origami: An Active Learning Exercise for Scrum Project Management

Christopher Sibona
Saba Pourreza
Stephen Hill
Department of Business Analytics, Information Systems and Supply Chain Management
University of North Carolina Wilmington
Wilmington, NC 28403, USA
sibonac@uncw.edu, pourrezas@uncw.edu, hills@uncw.edu

ABSTRACT

Scrum is a popular project management model for iterative delivery of software that subscribes to Agile principles. This paper describes an origami active learning exercise to teach the principles of Scrum in management information systems courses. The exercise shows students how Agile methods respond to changes in requirements during project implementation, one of the four Agile principles, in a deeper manner than many Agile active learning exercises. This learning activity uses an uncommon approach in Agile exercises in that tasks are provided, estimates made, progress is measured, and pivots to new tasks can be introduced based on task progress. All students were introduced to Scrum through two different lessons – one lecture-focused and one activity-focused. Students were surveyed after each lesson to determine lesson effectiveness. Students indicated they understood Agile concepts after completing the exercise and found the activity engaging. Students' perceptions of Agile were similar for both lecture and activity lessons. The results from the study find that students' perception of Agile learning increased when they had the lecture followed by the activity. If class time is constrained to a single lesson then the activity would be more beneficial than the lecture. Detailed instructions are provided for instructors to complete this activity.

Keywords: Agile, Scrum, Project management, Active learning

1. INTRODUCTION

The systems development life cycle (SDLC) is a standard topic in introductory systems analysis and design courses. For example, recent textbooks from Satzinger, Jackson, and Burd. (2016); Valacich, George, and Hoffer (2015); and Dennis, Wixom, and Tegarden (2015) cover the systems development life cycle and cover Agile methods specifically. The systems development life cycle provides approaches and methods for the development of a new information system. Satzinger, Jackson, and Burd (2016) and Dennis, Wixom, and Tegarden (2015) discuss the Agile method Scrum (Hirotaka and Ikujiro, 1986; Schwaber and Beedle, 2002) in particular as a popular method within the Agile method framework (Fowler and Highsmith, 2001). Agile methods are discussed to demonstrate how a system can be grown over time through iterations and incremental delivery of software and how to manage project risks (Satzinger, Jackson, and Burd, 2016). Larman and Basili (2003) note that iterative and incremental development methods have been used since the 1970s in some advanced development organizations, so this particular approach is not unique to Agile development. Stellman and Greene (2015) argue that Agile practices are more than the set of practices (e.g., iteration) but also a philosophy and a mindset.

Agile methods have a four-part philosophy focused on: individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan (Fowler and Highsmith, 2001). There are twelve principles behind the Agile Manifesto that clarify the development method. The Agile Manifesto clearly defines its supporting principles, like delivering software frequently, and the adoption of Agile development methods may have increased the use of iterative development by organizations. The topics of the systems development life cycle, Agile principles, and the Scrum method are of critical importance to the software development success.

The goal of this active learning exercise is to have students understand how Agile principles and the Scrum method, in particular, aid the delivery of working software with uncertain or changing requirements. The exercise described in this article is designed to help students understand the roles involved in the Scrum process, the purpose of the daily meeting, how iterations and incremental delivery of software work, how changing requirements are managed in Scrum, the purpose of the product and Scrum backlog, and the difficulty of project estimation and feature estimation.

The learning exercise is a structured, active learning activity that requires students to receive a list of requirements (origami diagrams) and materials (paper), develop estimates, break tasks into iterations, and have a daily scrum meeting for each simulated day. The activity is designed to have students understand how Scrum management practices may aid the delivery of software. It is important for students to understand why Scrum practices are particularly well-suited for software and in which contexts Agile practices will work well. The activity fits into a 50-minute academic class and can be adapted to longer classes by increasing the number of iterations. The activity is easy to learn, fun, and challenging.

The activity described in this article has a strong focus on the Agile philosophy element where responding to change is valued more than following a plan (Fowler and Highsmith, 2001). Students learn how Scrum manages changes in requirements in this learning activity, which is uncommon for Agile learning games. Students will estimate the tasks on their sprint backlog and then start their sprint. Students will then attempt to complete the tasks in the sprint backlog during the sprint iteration. At the end of the sprint, students show their progress to the product owner. Typically, only a minority of students will complete the tasks within their time estimate (despite student estimates that predicted the tasks would be complete within the sprint). At the end of the sprint, the product owner will assess the task and then introduce a change in the requirements. The change in requirements introduces the concept of pivoting where a "structured course correction" is made for business reasons (Ries, 2011) and is a major tenet in the Agile philosophy. The product owner states that the initial task is taking longer than expected and that a simpler task will be an adequate substitute. The more complex task is removed, a simpler and similar task is introduced to the next sprint backlog, and the sprint iteration is started. The activity described in this article will engage students more fully so they understand how Scrum effectively manages changes in requirements.

This study provides two contributions: the set of instructions for a Scrum activity that teaches students about Scrum and a critical evaluation of the experiment that compares the activity to a lecture on the same topic. The activity description consists of a detailed plan to guide an instructor through the lesson and breaks the lesson down into 13 tasks with time estimates and goals for each task. Scrum, as a course topic, was covered in two 50-minute lessons – one student group had the lecture followed by the activity and the other the activity followed by the lecture. Students completed a survey after each lesson to determine their perceptions of learning Agile content and lesson effectiveness and to evaluate their perceptions of Scrum. Students were asked three questions about each lesson to determine whether the lesson was engaging, imparted knowledge, and was relevant to the classroom topic. The two lessons were evaluated to measure perceptions of Agile learning and lesson effectiveness by method. The research questions examined in this study include:

- How effective was the Scrum activity and lecture?
- Are there differences in lesson effectiveness (engagement, relevancy, knowledge) for the lecture and activity?
- How should the lessons be ordered – lecture followed by activity or activity followed by lecture? Or does it matter?

The next section of this paper includes a literature review about the Agile games, active learning, and iterative development and is followed by a description of the activity, folding origami, as an active learning activity to understand Scrum project management. Finally, we evaluate the class lesson results after students completed the lecture and activity lessons on project management.

2. LITERATURE REVIEW

2.1 Agile and Scrum Learning Exercise

Active learning approaches have been used by educators both inside and outside the classroom to engage learners to understand Agile development principles and the Scrum method. Active learning approaches to learn Scrum methods have been used by practitioners and academics alike. For example, Tastycupcakes.org is a website dedicated to game development for consultants who teach in professional settings and focuses on Agile games. The site has over 150 submissions in the Agile category for its games. Several games for learning Scrum have been discussed in the academic and professional training area including: PlayScrum, Scrumia, and Scrum Simulation with LEGO Bricks (Table 1). The list of games is not intended to be complete but to offer a variety of different games for learning in different settings and their focus. None of the games in Table 1 vary the tasks during the game execution at the end of a sprint.

The activities focus on different concepts and require varying amounts of time to complete the activity. Some exercises focus on the product owner's role where choosing tasks from the product backlog to enter into a sprint backlog are the primary purpose of the learning activity. Other games focus on understanding the daily meeting, or iterative development. The games vary in quality and specification, and some may not fit well into academic course structures (Lee, 2016; Von Wangenheim, Savi, and Borgatto, 2013). Games for the classroom should be relatively quick to learn and play to be effective in the classroom as well as engaging and interactive (Baker, Navarro, and van der Hoek, 2005; Paasivaara et al., 2014). A challenge for classroom games is to have students reach the learning goal with minimal distractions from game mechanics, but ensuring that the game is complex enough such that the game reflects real-life situations (Baker, Navarro, and van der Hoeck, 2005).

Presenting Agile principles and the Scrum method in an educational setting in a way that promotes active student learning can be difficult for instructors. To combat this challenge, a variety of instructional exercises have been created that allow students hands-on opportunities to experience aspects of the systems development life cycle (Fernandes and Sousa, 2010; Lee, 2016; Paasivaara et al., 2014; Von Wangenheim, Savi, and Borgatto, 2013). Such active learning exercises have the added benefits of increased student engagement and learning (Auster and Wylie, 2006; Bonwell and Eison, 1991; Freeman et al., 2014; Prince, 2004). The activity described in this article adds to and enhances this body of classroom exercises. Specifically, the approach taken in this

Game	Type	Focus	Requirements Change
PlayScrum (Fernandes and Sousa, 2010)	Board & Card Game	Product backlog, sprint backlog, Scrum roles, sprint purpose, daily meeting	No
Scrumia (Von Wangenheim, Savi, and Borgatto, 2013)	Pen & Pencil & Paper Hats	Product backlog, sprint backlog, Scrum roles, meetings, concepts, task estimation, burn-down	No
Teaching Students Scrum using LEGO Blocks (Paasivaara et al., 2014)	Pen & Pencil & LEGO Blocks	Product backlog, sprint backlog, Scrum process and roles, requirements management, task estimation, team work, burndown	No
SCRUM-X (Lee, 2016)	Microsoft Excel	Product backlog, sprint backlog, estimation, prioritization, Scrum roles, Scrum process, requirements management, burndown	No
Ball Game (May, York, and Lending, 2016)	Ball Passing	Self-organizing team, Scrum framework, comparison to waterfall method, estimation	No

Table 1. Scrum Games

activity allows the product owner to change the tasks in the product backlog due to the progress made in the exercise. The exercise addresses Agile principle (4) “Responding to change over following a plan” (Fowler and Highsmith, 2001) where students can experience a change in requirements that varies from the initial plan due to their progress in the task completion.

2.2 Active Learning

Active learning methods are designed to engage students more during the instructional process and can be helpful in understanding how Scrum helps software teams deliver working software. Prince (2004) defines active learning “as any instructional method that engages students in the learning process. In short, active learning requires students to do meaningful learning activities and think about what they are doing” (Bonwell and Eison, 1991). Lee (2016) suggests that active learning can enhance decision making by analyzing different scenarios and the paths that projects can take. Simulation games in project management classes have been increasingly used in educational settings (Lee, 2016).

Almost every instructional method will positively affect learning outcomes; the approaches that can be taken will depend on many factors including the context of the learning objectives and the course (Prince, 2004). One way to encourage students to think about what they are learning is to create activities that more fully engage the student where student outcomes are greatly improved even when the amount of time spent on a given topic is the same (Prince, 2004). Prince (2004) suggests that active learning methods have generally favorable outcomes and notes, in particular, that active engagement has very strong positive results for student learning objectives. The Scrum activity described in this article requires that students actively engage in the Scrum process through the use of the roles of Scrum, estimation, daily meetings, product and sprint backlogs, etc.

2.3 Iterative and Incremental Delivery Methods, Agile, and Scrum

Iterative and incremental development as techniques for software development have been in use since the 1970s (Larman and Basili, 2003), although the widespread adoption of iterative and incremental project management may be more associated with Agile methods (Fowler and Highsmith, 2001). Agile is guided by four values and twelve principles and has many different implementations, including eXtreme Programming (XP) (Beck, 2000), Lean (Poppendieck and Poppendieck, 2003), and Scrum (Schwaber and Beedle, 2002). Scrum has three major roles in the process: product owner, scrum master, and development team (Schwaber and Beedle, 2002). The product owner manages the product backlog, the list of all requirements that are needed in the product. The scrum master ensures that a daily meeting is held and finds resources so that progress can be made by the development team. The development team is responsible for completing the tasks of the product backlog and sprint backlog. The scrum master is not a manager in that the scrum master does not tell people what to do, but acts as a facilitator so that the team can complete their tasks by ensuring the team follows the processes and shields the team from outside interference (Paasivaara et al., 2014; Schwaber and Beedle, 2002; Von Wangenheim, Savi, and Borgatto, 2013).

At the beginning of each sprint (iteration), the product owner and team meet to determine the priority of the tasks and what tasks will be entered into the sprint backlog that will be developed during the iteration. Each task (a use case or user story) is assigned to the sprint backlog and should be completed during the iteration. Once the sprint begins, team members complete the activities required to complete the task. Team members meet each day for approximately 15 minutes to individually answer three questions (Schwaber and Sutherland, 2013):

1. What have you done since the last daily Scrum (during the last 24 hours)?
2. What will you do by the next daily Scrum?
3. What kept you or is keeping you from completing your work?

At the end of a sprint, the team will hold an informal sprint review to demonstrate the progress to all stakeholders. The team will hold a sprint retrospective to review what went well during the sprint and improvements that can be made during the next sprint. The product owner will review the product backlog with the team and add new tasks to the sprint backlog in collaboration with the team members. The Scrum activity in this article requires students (acting as team members) to use iterative development to develop their product (origami). Daily meetings are held for each iteration day so that progress can be discussed among the team members with the Scrum master. If team members are unable to complete their work, they can ask for assistance.

3. EXERCISE AND DISUCSSION PLAN

The activity description includes the preparation of the exercise, the exercise execution, and the post-exercise debriefing. The activity and lecture lessons were designed for undergraduate business students. The content was covered in two days and with two different lesson types; one cohort had the activity followed by the lecture, and the other cohort had the lecture followed by the activity. A small amount of time is required to describe the background material for students who have the activity first (approximately five minutes) compared to those who have the lecture first (approximately two minutes).

The exercise was completed in 50 minutes where 33 minutes were consumed by the exercise itself. If more time is allowed, additional iterations of the exercise can be completed, or the time for a given day can be lengthened and the number of iterations can remain the same, or a combination of the two options can be used. Time estimates were developed through multiple pilot iterations of the exercise and then validated during the classroom execution with undergraduates. One instructor was sufficient to complete the exercise in the time described for a class of 40 students. It may be helpful to have additional people to help teams who say that they are blocked to help them make progress.

3.1 Exercise Preparation and Setup

The exercise is conducted with a packet of origami instructions for each student and origami paper. Instruction packets can be given to each student or one packet for each team. Instruction packets consisted of five origami diagrams that students use for estimating purposes and instructions. Origami paper is needed for each student in the class; students are expected to complete at least one figure and potentially two or three, so the upper bound would be three origami sheets for each student. Origami paper is helpful in this exercise because it is square and can have a colored and white side that aids students in following the instructions. The instructor needs an instruction packet and two additional origami diagrams that are not contained in the student packet.

The two courses that used the approach presented in this paper were not focused on software programming but focused on requirements gathering and analysis of software systems (information systems analysis) and an introductory course in management information systems. Students will work on origami projects that represent software features and requirements. Working with origami allowed students to understand Agile processes in software development even though the delivery of software is not the goal of the exercise. Rico and Sayani (2009) recommend that students be introduced to Agile practices before they take software engineering courses because students should become familiar with the process and how it works before they start the development of a system.

3.2 Exercise Execution

The exercise is conducted in multiple phases. Students need to form small groups, be given instruction packets and origami paper, have a brief introduction to the overall exercise, work on the exercise using Agile principles, and then be debriefed.

Students work in groups of three or four. Each student should receive an origami instruction packet and two or three sheets of origami paper. Students will be told that in Agile project management the product backlog is a set of features to be completed and that the origami instruction packet of the five diagrams is the complete set of features for a given product.

Students will be introduced to the three roles in the Scrum method (Schwaber and Sutherland, 2013): product owner, scrum master, and team member. Product owners represent the client role and have additional responsibilities such as priority setting of the features. Product owners generally have wide latitude for deciding the priority of a given feature, but input from the development team may also be solicited because infrastructure and dependencies can impact the delivery of a future feature. For a feature to be started and completed it must first be placed on the product backlog. The product owner decides which features need to be implemented and the priority of the features, and the team members implement those features. The scrum master enforces Scrum practices to coordinate meetings, plans subsequent meetings to resolve issues, and holds a daily Scrum. The daily Scrum meeting is a meeting with all team members where members report their progress and plans for the next day. When the daily meeting is held, the scrum master asks each member of the team three questions about their progress, goals for the next day, and whether they are blocked (Section 2.3).

The role of scrum master can rotate through the team members. The classroom groups are asked to assign one person in their group as the scrum master who will ask each team member these questions during the daily Scrum.

Task 1: Introduce active learning exercise and pass out instruction packets and origami paper.

Task 2: To compare and contrast Agile project management to traditional waterfall project management, the groups will estimate how long it will take to complete the entire packet. Students will be given two minutes to provide an estimate. This task is outside normal Agile project management but is given to compare how estimating large projects is difficult and that estimating smaller tasks is both more accurate and easier. An analogy is provided to the students that it is fairly typical that management asks for an estimate of how long it will take to complete a large project.

Task 3: The students are instructed that a day in this exercise is five minutes, and that the iteration length is two days. A more typical recommendation is that an iteration is two to six weeks with a preference for shorter timescales (Fowler and Highsmith, 2001). The goal of dividing days into a five minute period and a sprint (iteration) of two days is to have learners experience a day and an iteration in a reasonably quick manner due to the time constraints of a classroom exercise. Students are instructed that an effort estimate is generated for how much effort a task (origami design) will take to complete. The effort estimate is often provided as a number in the Fibonacci sequence (1, 2, 3, 5, 8, 13, etc.) where estimating larger projects is considered difficult and the amount of error is larger. The goal in estimation should be to determine what may be done with a given constraint (sprint length) and that precision is not expected.

Task 4: The students' instruction packet contains two high priority designs to be implemented: the crane is the first priority and pigeon 1 is the second priority. They will estimate the effort for each design in minutes. The estimate must conform to a Fibonacci sequence number. The students are advised that if they estimate a design may take more than 10 minutes (sprint length) then they should change the scope of the task to indicate how far they will get in the iteration. In this example, a student could estimate that they will get to step six of a design in the iteration. In this way, the iteration length stays constant (typical in an Agile project), but the content can be reduced appropriately.

Task 5: Groups now have a goal for the iteration (Task 2) and will hold their daily Scrum meeting. The scrum master will ask each member what they plan to do in the next day. Each member indicates how far they will get in their design. If a student predicts they will complete the first design in five minutes and start the next task, then they will indicate that they will complete the design and how far they will get on the second design. The scrum master should record each answer from the team member.

Task 6: A five-minute timer is started, and the team members start working on their design. At the end of five minutes, the timer will sound and the team members are told that the day is over and to stop working on their design.

Task 7: Group members meet for their second Scrum as coordinated by their scrum master. They each answer the daily Scrum meeting questions as asked by the scrum master. The scrum master will record how far the team member went in the five minutes (which step, or completed design) and then ask how far they will get in the next iteration. If a team member is blocked (third question in the daily Scrum questions), then another team member may be directed by the scrum master to help the member who cannot complete the task. The classroom instructor may also be called to help, or any individual assistants who are present may aid the team member who is blocked from making progress.

Task 8: A five-minute timer is started, and the team members start working on their design. At the end of five minutes, the timer will sound and the team members are told that the day is over and to stop working on their design. The two day sprint is complete.

Task 9: The instructor will ask all students to show their progress (Agile methodologies measure progress through "working software" (Fowler and Highsmith, 2001) and, in this exercise, the origami is a measure of the "working software"). The instructor will then remove the first design in the origami instruction set (the one that most students will be working on) and replace it with a simpler task (pigeon 2). The concept of welcoming changing requirements (the second Agile principle) is introduced here. Changes happen, through no fault of the team members, product owners, or scrum master, due to several factors. Pigeon 2 is removed from the product backlog, and two new origami designs are introduced to the backlog.

Task 10: Students hold their third and final Scrum meeting with the final design.

Task 11: A five-minute timer is started, and the team members start working on their design. When the time is over, students are asked to show their designs.

Task 12: The students are then asked to reflect on the process and their progress and asked how they may be more effective in future iterations (Agile principle 12: At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly).

Task 13: Student debrief. Ask students whether it was easier to estimate the entire packet as a whole (task 2) or in smaller increments (Scrum approach). Review: roles, role responsibility, daily questions, estimation, product owner actions at the end of an iteration, product backlog, and sprint backlog. Higher-order questions are asked: Did stating what you will do in the next day motivate you? What actions can be taken if you are blocked? Does Scrum aid in transparency for those involved in the project, and if so, how? The exercise is now over.

Table 2 summarizes the tasks.

Task	Time (min)	Summary
1	5	Introduction to activity – hand out instruction packets and paper
2	2	Estimate entire packet
3	5	Exercise description and instruction: day length, iteration length, estimation parameters, product backlog
4	2	Product backlog priorities identified (Crane and Pigeon 1), estimate iteration progress
5	2	First Scrum meeting
6	5	First day started (5 minutes of work) – work on Crane, and, if complete, Pigeon 1
7	2	Second Scrum meeting
8	5	Second day started (5 minutes of work)
9	5	Iteration 1 complete – product owner (instructor) reviews work, decides to change requirements – Pigeon 2 is the new priority followed by the whale – drop Crane and Pigeon 1 designs – add new origami designs: water bomb and butterfly
10	2	Third Scrum meeting
11	5	Third day started (5 minutes of work) – Pigeon 2
12	5	Show progress – reflect on process
13	5	Debrief
	50	**Exercise Complete – Total Time**

Table 2. Task Summary

Question Number	ID	Question
1	Understand	I understand Agile project management
2	Importance	I understand the importance of Agile project management
3	Changes	I understand how Agile project management handles changes in requirements
4	Daily	I understand how having a "daily" meeting can aid in project management
5	Iterations	I understand how the use of iterations can aid in project management
6	Estimate-small	I understand how estimating smaller parts of a project can aid in project management
7	Estimate-large	I understand how estimating large projects is difficult
8	Transparency	I understand how Agile project management provides transparency to stakeholders throughout the project.
9	Primary	I understand how focusing on product delivery can be the primary measure of progress
10	Simplicity	I understand how simplicity – the art of maximizing the amount of work not done – is essential
11	Reflection	I understand how a team can reflect on the process after product delivery to become more effective and adjust its behavior for future projects
12	Engage	This exercise/lecture was engaging
13	Knowledge	I gained knowledge from this exercise/lecture
14	Relevant	This exercise/lecture was irrelevant for this course (reverse coded)

Table 3. Survey Questions

3.3 Evaluation of Student Exercise Participation and Performance

The origami was not graded on quality or quantity; the purpose of this experiment is to evaluate the two instructional methods used to teach Scrum processes. Students filled out a survey that asked several questions related to the Scrum method (Table 3). There were five sections of the information systems classes who completed this activity. Two sections had the activity followed by the lecture (AL), and three sections had the lecture followed by the activity (LA).

The questions have two groupings: the student's perception of learning Agile project management and lesson effectiveness. The assessment tool was based on a similar tool used by Pinder (2013a, 2013b, 2014), student participation was voluntary, and student responses were anonymized.

4. RESULTS

4.1 Lesson Analysis

The evaluation of the activity and lecture were conducted for the first and second lessons and differentiated by cohort (activity followed by lecture (AL) or lecture followed by activity (LA)). An online survey was used to collect the data for this study. Surveys from 155 students were collected over 2 semesters from 5 classes. Data was collected from three senior level information systems analysis classes and two introduction to management information systems classes. There are no programming class prerequisites for either class, although the majority of students in the information systems analysis class have had programming experience and will be required to have at least two programming classes before degree completion. All sections were taught by the same instructor.

Surveys were given to students after both the first and second lessons. Of the 155 students participating in the study, only 125 completed both surveys which resulted in dropping the 30 students who completed just one of the two surveys. The final participant count across the conditions was 41 students in the AL group and 84 students in the LA group. Students were also asked three open-ended questions: (1) What was the best part about this exercise/lecture? (2) What can be improved in this exercise/lecture? and (3) What did you learn from this exercise/lecture?

Classrooms were randomly assigned to one of two experimental conditions for this study – either an activity followed by lecture or lecture followed by activity order. A comparison of perceptual Agile learning outcomes after the first session is shown in Figure 1. Statistical differences are measured using Fisher's Exact test to determine whether student perceptions differ based on lesson type (Table 4). There is one statistically significant difference between the two cohorts – students in the active learning session had higher levels of agreement that estimating large projects was difficult. All other perceptual measures of Agile learning after the first lesson had no statistically significant differences. Beyond the Agile learning objectives, the students rated lesson effectiveness on engagement, relevancy, and knowledge. The activity was considered more engaging to students at statistically significant levels compared to the lecture (Table 4 and Figure 2). Students' perceptions of the knowledge and relevancy in both sessions were not statistically different.

A comparison of perceptual Agile learning outcomes after the second session is shown in Figure 3. Statistical differences are measured using Fisher's Exact test to determine if student perceptions differ based on lesson type (Table 5). There are four statistically significant differences between the two cohorts. Students who had the lecture first followed by the activity had higher levels of agreement for perceptions of understanding Agile project management, the importance of the daily meeting, understanding how estimating smaller parts of a project can aid in project management, and understanding how iterations can aid in project management. All other perceptual measures of Agile learning after the second lesson had no statistically significant differences. Once again, the activity was considered more engaging to students (Table 5 and Figure 4). Students perceptions of the knowledge and relevancy in both lessons were not statistically different.

An alternative perspective is provided in Tables 4 & 5 where the item values are treated as interval values to generate means, mean differences, and statistical differences. Mann-Whitney *U* test was used to calculate statistical significance.

	Ordinal	Interval Treatment					
Agile Concept	p-value	Mean	Std. Dev.	AL Group	LA Group	Cohort Mean Difference	p-value
Change	0.2956	5.5118	1.1399	5.4545	5.4940	0.0515	0.5623
Daily Meeting	0.3880	5.8661	1.0417	5.8864	5.8554	0.0309	0.6007
Estimating (small)	0.7355	5.7165	0.9419	5.8409	5.6506	0.1903	0.2315
Estimating (large)	0.0177 *	5.9843	1.0763	6.2727	5.8313	0.4414	0.0027 **
Importance	0.8922	5.6063	1.0776	5.6591	5.5783	0.0801	0.6365
Iterations	0.3215	5.4173	1.2998	5.6364	5.3012	0.3352	0.0467 *
Reflection	0.1811	5.6535	1.0644	5.7500	5.6024	0.1476	0.2352
Simplicity	0.2593	5.2992	1.4547	5.4318	5.2289	0.2029	0.3440
Software Focus	0.7845	5.3465	1.1154	5.5682	5.2289	0.3393	0.1082
Transparency	0.8843	5.4016	1.4101	5.5227	5.3373	0.1854	0.3737
Understand	0.2705	5.2992	1.0934	5.4545	5.2169	0.2377	0.1726
Engagement	0.0001 ***	5.0630	1.5723	5.8409	4.6506	1.1903	0.0001 ***
Irrelevant	0.1848	5.1811	1.6970	5.0227	5.2651	-0.2424	0.4447
Knowledge	0.4551	5.3858	1.1889	5.7045	5.2169	0.4877	0.0261 *

Table 4. Perceptions of Learning and Lesson Effectiveness – First Session

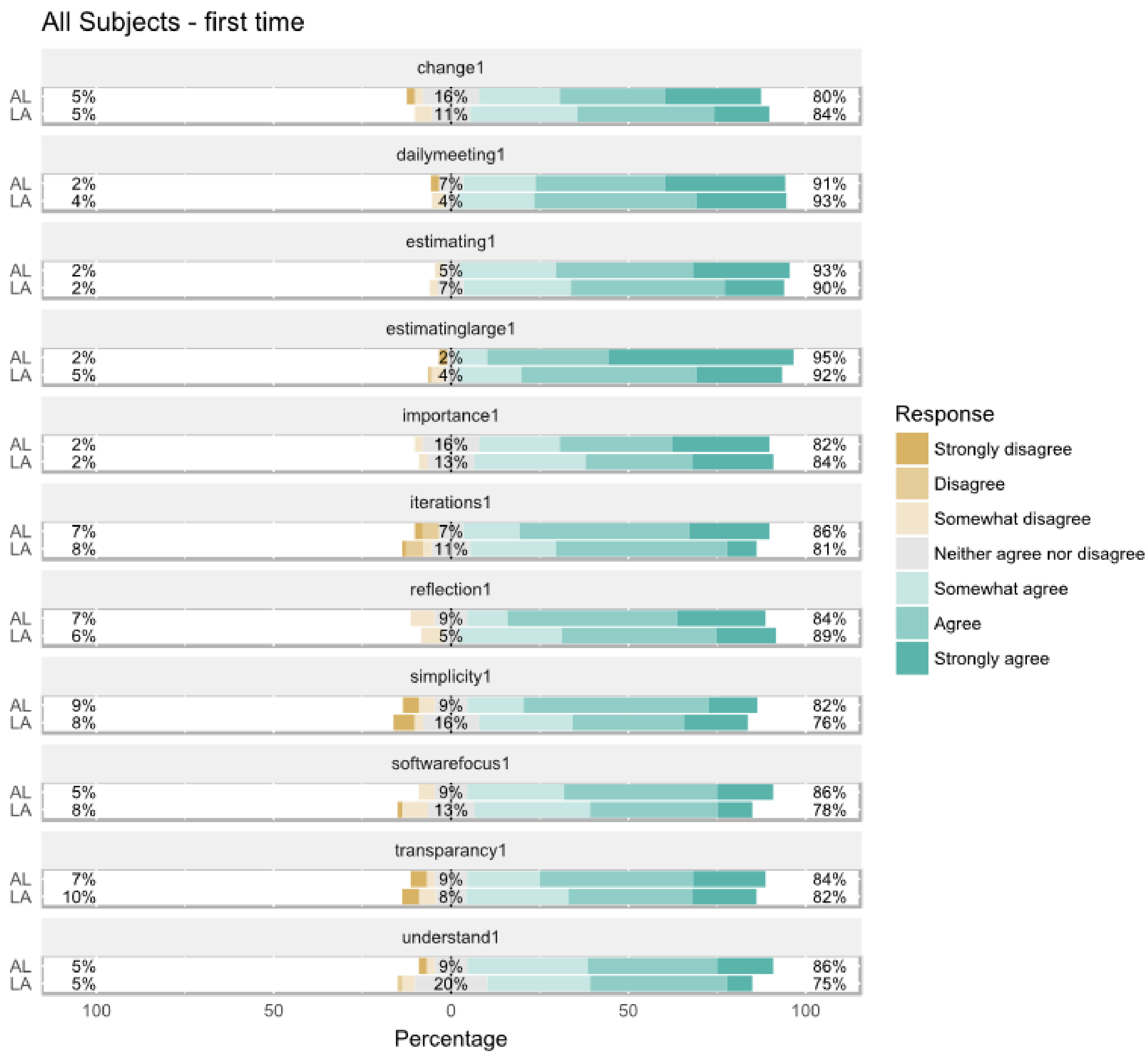

Figure 1. Perceptions of Agile Learning - First Session

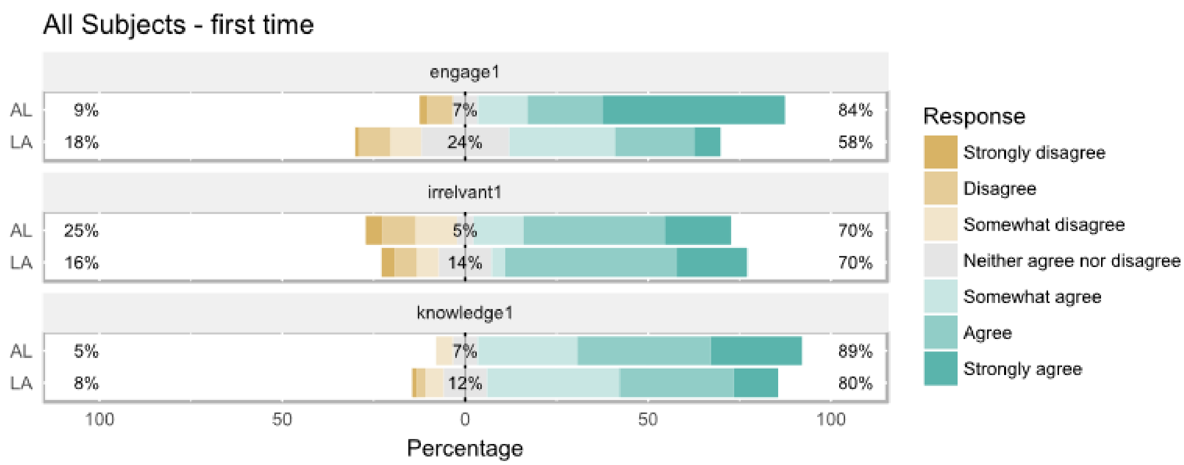

Figure 2. Lesson Effectiveness - First Session

	Ordinal	**Interval Treatment**					
Agile Concept	**p-value**	**Mean**	**Std. Dev.**	**AL Group**	**LA Group**	**Cohort Mean Difference**	**p-value**
Change	0.0081 **	5.9457	0.7837	5.6512	6.0930	-0.4419	0.0020 **
Daily Meeting	0.0409 *	6.1163	0.8894	6.0233	6.1628	-0.1395	0.7214
Estimating (small)	0.0328 *	6.1240	0.7604	5.9070	6.2326	-0.3256	0.0619
Estimating (large)	0.3610	6.2636	0.7450	6.1628	6.3140	-0.1512	0.1981
Importance	0.0917	5.8527	0.8669	5.5814	5.9884	-0.4070	0.0153 *
Iterations	0.0049 **	5.8760	0.9015	5.5116	6.0581	-0.5465	0.0010 ***
Reflection	0.0672	6.0465	0.7589	5.8837	6.1279	-0.2442	0.1427
Simplicity	0.1561	5.8217	0.9474	5.7209	5.8721	-0.1512	0.1267
Software Focus	0.2899	5.8217	0.8790	5.6279	5.9186	-0.2907	0.0813
Transparency	0.2203	5.6667	1.0777	5.4651	5.7674	-0.3023	0.0821
Understand	0.1406	5.8605	0.8454	5.6279	5.9767	-0.3488	0.0255 *
Engagement	0.0001 ***	5.9225	1.2094	5.1860	6.2907	-1.1047	0.0001 ***
Irrelevant	0.7010	4.9845	1.9684	5.3023	4.8256	0.4767	0.2908
Knowledge	0.2216	5.9070	1.0034	5.6744	6.0233	-0.3489	0.0831

Table 5. Perceptions of Learning and Lesson Effectiveness – Second Session

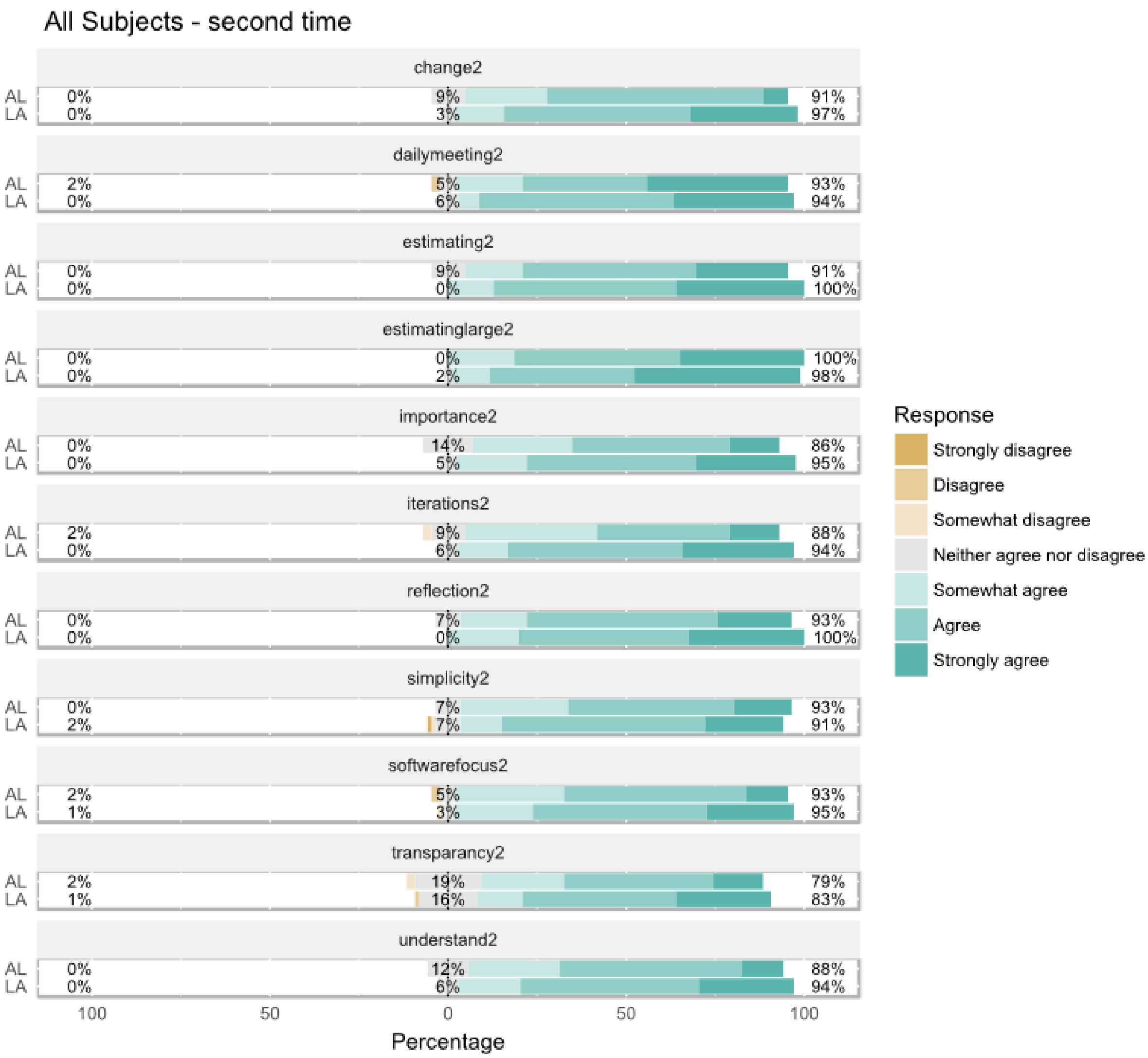

Figure 3. Perceptions of Agile Learning - Second Session

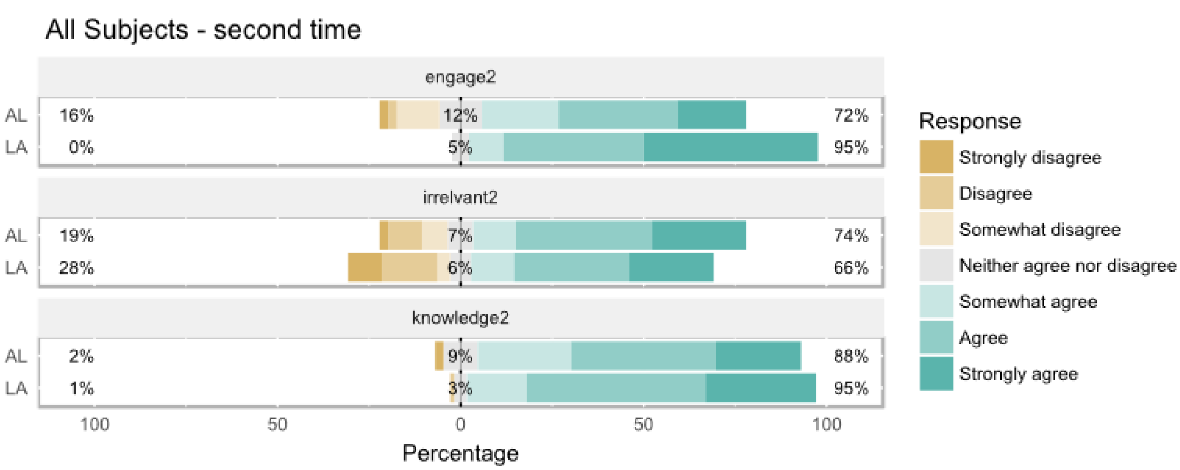

Figure 4. Lesson Effectiveness - Second Session

Agile Concept	p-value all students	Sig.	p-value AL Group	Sig.	p-Value LA Group	Sig.
Change	0.0011	**	0.0142	*	0.0009	***
Daily Meeting	0.0415	*	0.9655		0.0081	**
Estimating (small)	0.0024	**	0.5131		0.0001	***
Estimating (large)	0.2921		0.1653		0.0129	**
Importance	0.0307	*	0.3967		0.0176	**
Iterations	0.0185	*	0.1143		0.0004	***
Reflection	0.0069	**	0.4626		0.0028	**
Simplicity	0.0185	*	0.3405		0.0021	**
Software Focus	0.0025	**	0.6219		0.0014	**
Transparency	0.0054	**	0.6210		0.0023	**
Understand	0.0007	***	0.5682		0.0001	***

Table 6. Fisher's Exact Test – Increases in Perceptions of Learning for Second Lesson

4.2 Perceptions of the Second Lesson Analysis

Additional analysis was performed to determine how the lessons should be ordered. The analysis examined whether students (regardless of cohort) perceived increases in knowledge between lesson 1 and lesson 2 and found that ten of the eleven perceptual measures showed an increase in levels of agreement that the concepts of Agile were better understood after the second lesson (Table 6). That is, student perceptions of their understanding of Agile increased with the second lesson for ten of eleven measures. Additional analysis was performed to determine whether the cohort of students (AL or LA) had statistically significant changes in their perceptions of knowledge of Agile between lesson 1 and lesson 2. In the activity followed by lecture group, only one of eleven measures was statistically significantly different – students in the AL group said they understood how Agile project management handles changes in requirements at statistically significantly higher levels. In the lecture followed by activity group, all eleven of the Agile perceptual knowledge questions had statistically significant increases in their level of agreement. The activity was consistently ranked higher in terms of engagement compared to the lecture regardless of whether the activity was delivered first or last.

4.3 Open-Ended Question Response Summary

Three open-ended questions about the exercise and lecture were asked to students to discover the best part of the exercise, what can be improved, and what students learned. Students indicated that the best part of the exercise was that the active learning was engaging, taught an important lesson, was applied, and students learned about daily Scrum meetings. Student suggested the following improvements: the task be more business-oriented, more time dedicated to the origami, increase the number of iterations, origami was frustrating and hard (the same student noted that that the difficulty was intentional), more focus on the roles in Scrum, and to incorporate the lecture and the exercise more tightly. Students said they learned about Agile project management, its practices and importance, that making your own estimates for a task will drive workers to complete a goal, how Agile can be both annoying and necessary, the importance of daily meetings to understand the project status, how to estimate a job and make adjustments, how iterations can allow product owners to pivot to new tasks if sufficient progress is not being made, how predicting task times is difficult, the importance of iterations and estimation, how to set realistic goals for a project, and that holding daily meetings is important and more efficient. This summary collection is not a complete set of the helpful comments but provides some insight into the student experience of the exercise.

5. DISCUSSION

The purpose of this study was to develop a set of instructions for a Scrum activity and to evaluate the experiment to determine the lesson effectiveness. The instructions presented here cover many aspects of project management and pay particular attention to how Scrum manages changing requirements, which is relatively uncommon in Scrum exercises. The experiment was conducted to determine the answer to three research questions: 1) How effective was the Scrum activity and lecture? 2) Are there differences in lesson effectiveness (engagement, relevancy, knowledge) for the lecture and activity? and 3) How should the lessons be ordered – lecture followed by activity or activity followed by lecture? Or does it matter?

Student perceptions of both the lecture and the activity are largely positive. Students had high levels of agreement of their perceptions of learning the material after the first lesson, regardless of whether the student had the activity or the lecture. Students found the activity to be more engaging than the lecture, but did not find the relevancy or the knowledge to be statistically significantly different. One difference was found in the first lecture between the two cohorts. Those who had the activity first said they understood that estimating large projects was difficult at higher levels of agreement compared to the lecture cohort. Students were asked to estimate the entire packet of five origami figures, and many students (but not all) estimated that they could complete the entire packet within the fifty minute time frame. The activity cohort could then compare their actual progress completed throughout the activity and discover that they (often) overestimated their ability to complete the task. The lecture group was told that these initial estimates are often inaccurate but did not have a recent experience with their estimates and the subsequent confirmation so they had no results to which they could readily compare. In general, though, the students' perceptions of these

Agile concepts are relatively similar and high regardless of the lesson experience.

Student perceptions of the second lesson varied more than the first, where the lecture followed by activity cohort indicated higher levels of agreement in four areas of their perceptions in Agile concepts. The first lesson provided a foundation for the second lesson regardless of lesson ordering; however, the lecture followed by activity group had higher levels of agreement that they understood how Agile handles changes in requirements, the purpose of the daily meeting, how estimating small tasks is easier than large, and how iterations help with project management. One notable difference is that both groups had high levels of agreement that estimating large projects is difficult. Again, the students found the activity more engaging than the lecture.

To answer the third research question on lesson ordering, we compare the results from lesson one and lesson two both with and without group cohorts. Lesson one is the first lesson in time regardless of the lesson type (A or L), where lesson two is the second lesson of the two lesson set (L or A). The results of the analysis of the second lesson with and without cohort groupings is somewhat surprising given that the overall levels of agreement from the second lesson vary in four of eleven items (Table 6). There is more growth for the students in the lecture followed by activity group for the second lesson compared to the activity followed by lecture (Table 6). Based on the results of this study, the recommendation would be to have two lessons where the first lesson is the lecture followed by the activity. The students in the LA group perceived that they learned more compared to the AL group after the second lesson. If time permits only one lesson, then the activity should be prioritized because the activity was more engaging and the increases in the perceptions of Agile management for the second lesson were much lower compared to the lecture. The higher levels of agreement for the LA group on all eleven perceptual measures may be that their experience with the activity meaningfully deepened their learning. The students may have thought they understood the concepts after the first lesson so the levels of agreement in the two groups (regardless of the lesson type) were high (Table 4). The second lesson may have been more meaningful and allowed the students to think about what they were doing because they were prepared by the lecture (Bonwell and Eison, 1991) so their perceptual measures may have increased more strongly compared to the other cohort. The activity and lecture both appear to have increased student perceptions of how Agile project management aids software delivery.

6. CONCLUSIONS

This article describes an active learning exercise for Scrum project management for students to understand the roles involved in the scrum process, the purpose of the daily meeting, how iterations and incremental delivery of software work, how changing requirements are managed in Scrum, the purpose of the product and scrum backlog, and how to estimate an entire product and estimate a feature. The activity described in detail in this article uses origami to represent features in software development. The activity was successfully tested in two different courses, an introductory management information systems course and an advanced information systems analysis course. Students were able to directly experience how Scrum incorporates a product backlog, a sprint backlog, changing requirements, estimation, and daily scrum meetings through origami. Introducing an active learning component for Scrum is likely beneficial (Bonwell and Eison, 1991; Prince, 2004).

The analysis indicates that both the activity and lecture were perceived positively by students. As expected, students found the activity to be more engaging than the lecture. There are only small differences in student perceptions of relevancy and knowledge when the activity and lecture are compared. The lesson ordering analysis indicates 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, and there are only small differences in relevancy and knowledge lesson effectiveness measures.

This exercise helped demonstrate the importance of Scrum so that students can gain a deeper understanding of Scrum as a system – not only how the process works but why the process works. A student in the advanced information systems analysis class stated that the lessons helped him gain an appreciation and deeper understanding for project management. This student had a summer internship in systems development and used Scrum, but said he never really understood why it was being used. The student said he understood the purpose of Scrum after the origami lesson. Students appeared to benefit from both the lecture and activity lessons as taught in both classes as their level of agreement went up on the perceptual measures of Agile after the second lesson regardless of the cohort (activity followed by lecture or lecture followed by activity). Interest in Agile project management and Scrum's implementation of Agile principles appears to be growing, and the lesson described here will help students gain a deeper understanding of Agile principles.

7. REFERENCES

Auster, E. R. & Wylie, K. K. (2006). Creating Active Learning in the Classroom: A Systematic Approach. *Journal of Management Education*, 30(2), 333–353.

Baker, A., Navarro, E. O., & van der Hoek, A. (2005). An Experimental Card Game for Teaching Software Engineering Processes. *Journal of Systems and Software*, 75(1-2), 3–16.

Beck, K. (2000). *Extreme Programming Explained: Embrace Change*. Addison Wesley Professional.

Bonwell, C. C. & Eison, J. A. (1991). Active Learning: Creating Excitement in the Classroom. *1991 ASHE-ERIC Higher Education Reports*. ERIC.

Dennis, A., Wixom, B. H., & Tegarden, D. (2015). *Systems Analysis and Design: An Object-Oriented Approach with UML (6e)*. John Wiley & Sons.

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

Fowler, M. & Highsmith, J. (2001). The Agile Manifesto. *Software Development*, 9(8), 28–35.

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active Learning Increases Student Performance in Science, Engineering, and Mathematics. *Proceedings of the National Academy of Sciences*, 111(23), 8410–8415.

Hirotaka, T. & Ikujiro, N. (1986). The New Product Development Game. *Harvard Business Review*, 64(1), 137–146.

Larman, C. & Basili, V. R. (2003). Iterative and Incremental Developments. A Brief History. *Computer*, 36(6), 47–56.

Lee, W. L. (2016). Scrum-X: An Interactive and Experiential Learning Platform for Teaching Scrum. The 7th International Conference on Education, Training and Informatics (ICETI 2016), Orlando, FL.

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

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

Pinder, J. P. (2013a). An Active Learning Exercise for Introducing Agent-based Modeling. *Decision Sciences Journal of Innovative Education*, 11(3), 221–232.

Pinder, J. P. (2013b). An Excel Solver Exercise to Introduce Nonlinear Regression. *Decision Sciences Journal of Innovative Education*, 11(3), 263–278.

Pinder, J. P. (2014). A Demonstration of Regression False Positive Selection in Data mining. *Decision Sciences Journal of Innovative Education*, 12(3), 199–217.

Poppendieck, M. & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit*. Addison-Wesley.

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

Rico, D. F. & Sayani, H. H. (2009). Use of Agile Methods in Software Engineering Education. *In Proceedings of the 2009 Agile Conference*, AGILE '09, 174–179, Washington, D.C.

Ries, E. (2011). *The Lean Startup: How Today's Entrepreneurs use Continuous Innovation to Create Radically Successful Businesses*. Crown Business.

Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2016). *Systems Analysis and Design in a Changing World (7e)*. Cengage Learning.

Schwaber, K. & Beedle, M. (2002). *Agile Software Development with Scrum, volume 1*. Prentice Hall Upper Saddle River.

Schwaber, K. & Sutherland, J. (2013). *The Definitive Guide to Scrum: The Rules of the Game*. Scrum.Org and Scrum Inc.

Stellman, A. & Greene, J. (2014). *Learning Agile: Understanding Scrum, XP, Lean, and Kanban*. O'Reilly Media, Inc.

Valacich, J. S., George, J. F., & Hoffer, J. A. (2015). *Essentials of Systems Analysis and Design (7e)*. Pearson Education.

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.

AUTHOR BIOGRAPHIES

Christopher Sibona is an Assistant Professor of Management

Information Systems at the University of North Carolina – Wilmington where he teaches undergraduate courses in information systems analysis. His research interests include social media, continuance intention, and project management

Saba Pourreza is an Assistant Professor of Supply Chain at the

University of North Carolina – Wilmington. Her research lies in the area of life cycle affordability of a product and performance-based logistics. She teaches Supply Chain Management and Purchasing & Material Management at UNCW.

Stephen Hill is an Associate Professor of Business Analytics

in the Business Analytics, Information Systems, and Supply Chain Department in the Cameron School of Business at the University of North Carolina – Wilmington. A native of Alabama, Dr. Hill received his doctorate in Operations Management from the University of Alabama in 2008. At UNCW, Dr. Hill teaches courses in analytics, operations management, and supply chain management. He has received several awards in recognition of his teaching. Dr. Hill's research focuses on application of analytics techniques in a variety of areas, such as supply chains, sports, and healthcare. He is also interested in research related to innovative approaches to education. Dr. Hill enjoys traveling, photography, and sports (especially college football).

Information Systems & Computing
Academic Professionals

Serving Information Systems Educators

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