Improving Content Learning and Student Perceptions in CS1 with Scrumage

Shannon Duvall
Computer Science
Elon University
Elon, NC USA
sduvall2@elon.edu

Scott Spurlock
Computer Science
Elon University
Elon, NC USA
sspurlock@elon.edu

Dugald Hutchings
Computer Science
Elon University
Elon, NC USA
dhutchings@elon.edu

Robert Duvall
Computer Science
Duke University
Durham, NC USA
rcd@cs.duke.edu

ABSTRACT

Scrumage (SCRUM for AGile Education) is a recently proposed classroom management technique in which students are given autonomy to choose individually from a variety of pedagogies (e.g., traditional lectures, active learning, a flipped-based approach, etc.). The result is multiple simultaneous pedagogical styles in a single course. In this paper we present the results of comparing six sections of an introductory programming course at the same university, three of which used Scrumage and three of which took a traditional approach. We administered surveys of both content acquisition and learning attitudes at the beginning and end of the course. While students in all sections improved in content learning, the students in the Scrumage classrooms outperformed those in the traditional sections. The improvement in content learning was also more uniformly distributed among students, not limited to the high achievers. Scrumage students showed generally improved attitudes about learning after the course, especially in the areas of Effort Regulation (perseverance in problem solving) and Control of Learning (taking responsibility for learning success). We observed some correlation between this metalearning and improvements in content scores in the Scrumage sections, but not in the traditional sections. Finally, based on an analysis of student comments in the Scrumage sections, we show that as the course progressed, positive student comments about their abilities and confidence were more common, even as course material became more difficult. We believe that positive attitude changes we saw with Scrumage mean it has potential for widening the retention of students in Computer Science.

CCS CONCEPTS

• Social and professional topics~Computing education • Social and professional topics~Computer Science education Social and professional topics~CS1

SIGCSE '21, March 13–20, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8062-1/21/03…$15.00
https://doi.org/10.1145/3408877.3432415

KEYWORDS

Computing Education; Classroom Management; Pedagogy; Scrum

ACM Reference format:

Shannon Duvall, Scott Spurlock, Dugald Hutchings, and Robert Duvall. 2021. Improving Content Learning and Student Perceptions in CS1 with Scrumage. In *Proceedings of The 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13-20, 2021, Virtual Event, USA.* ACM, New York, NY. 7 pages. DOI https://doi.org/10.1145/3408877.3432415

1 INTRODUCTION AND RELATED WORK

There has long been discussion on the best way to impart information in a higher educational setting. Proponents of the "sage on the stage" [2] prefer lectures as an efficient means of relaying large amounts of information quickly and easily. Others prefer the professor to be a "guide on the side" [10] so that students take a more active role during the learning process. There is no consensus on a pedagogy style that leads to better learning outcomes for everyone; rather it seems to be a personal preference for the teacher or learner. While the research on the importance of these preferences to content learning is inconclusive, there is evidence that various pedagogy techniques affect other measures like student attitude, motivation, and a sense of belonging [7,8,22,23]. Further, positive student perceptions have been linked to resilience, engagement, and continuing in the field of study [5,20].

Unfortunately, what is helpful to one group of students can be off-putting to another group. For example, using peer instruction or team learning has been found to be helpful for promoting persistence in underrepresented racial groups [11] as well as being welcoming to women [1], but these practices can be disengaging or overwhelming for people with certain disabilities, such as some autism spectrum disorders [6,16]. As Rose et al. concludes, "There is no one means of engaging students that will be optimal across the diversity that exists" [16]. Employing multiple methods of engaging students not only aids members of underrepresented groups, but generally helps all students connect with the material [15,17,18].

The novel pedagogy method called Scrumage [3,4] (SCRUM for AGile Education) aims at giving students the power to choose their own pedagogy style, despite the fact that other students in the same course may be choosing differently. It engages the agile methods of Scrum [21] to let students change their preferred learning method as the course continues. In Scrumage (as in Scrum) work takes place in short bursts called sprints, and self-regulating teams prioritize tasks to schedule the completion of work. At the end of each sprint, the work is assessed by the client (or professor, in this case) and the teams reflect on what methods were and were not effective and efficient. The Scrum methodology is widely adopted as a successful project management technique largely because it emphasizes lean processes (no busy-work), autonomous teams (no dictated decisions from management), and a fast feedback loop on both process and product (no static plans).

In Scrumage, the work being undertaken is the learning of course objectives; the client is the instructor. The completion of deliverable projects like assignments are a byproduct of the main "project" of mastering the material at hand. To mirror the values of Scrum, Scrumage emphasizes allowing students to have as much control over course management choices as possible and allowing them to try new methods as the course progresses based on results. Students are generally allowed to work in teams whose size and processes are largely unregulated by the professor. Importantly, students get to choose individually how they use class time by making requests from the professor. Learning methods like reading the textbook, listening to a lecture, watching a video, or completing a worksheet are not generally dictated, required activities but rather comprise a menu of options from which students can choose. In this way, students avoid activities they believe to be ineffective for themselves personally and are more committed and engaged in the activities they do choose. At the end of the unit, or sprint, students are assessed in usual ways with assignments, quizzes, or tests. They then complete a retrospective in which they reflect on their success – both in how they performed grade-wise and also as to which learning activities were most effective for them. They can then make informed decisions on how to learn in the next sprint.

Prior work describing Scrumage implementations in Discrete Math [4] and Algorithm Analysis [19] courses found that using the Scrumage technique resulted in students taking more responsibility for their learning, better perception of the course material and its importance, and overall improved attitudes. Follow-on work [3] found that Scrumage student grades were higher than in a traditional course and provided further anecdotal evidence of metacognitive learning and character building. The authors also suggest best practices for running a course with Scrumage.

In this work we report results from implementing Scrumage in several sections of an introductory programming course and comparing these results to sections run in a traditional manner. We present data gathered on student attitudes and learning from both types of course using pre-term and post-term surveys. We also discuss the variations made in the original Scrumage methodology, as we tailored the technique to fit our course and professors.

2 STUDY DESIGN

Whereas previous research examined the impact of Scrumage independent of other methods, here we endeavor to examine as directly as possible differences in outcomes between sections of a course taught with Scrumage and sections of a course taught in a more typical fashion. Specifically, we aim to address three areas:

1. differences, if any, in *changes in learning attitudes*;
2. differences, if any, in *knowledge/content acquisition*;
3. relationships, if any, between these characteristics.

2.1 Courses

We examined six sections of an introductory computer science course (CS1) in the August-December 2018 term of a liberal arts institution in the southeastern United States. The enrollments of each of the sections were between 16 and 28 students. The sections met either twice a week for 100 minutes or thrice a week for 70 minutes, totaling about 3½ hours per week of in-class time. All classrooms were equipped with desktop computers and wireless Internet access, with students permitted to use personal laptops or other equipment for coursework.

Three of these six sections were delivered using Scrumage. One instructor taught two sections and another instructor taught one section. Both instructors had experience teaching using the Scrumage technique. In all sections, the textbook *Big Java* [9] was listed as a recommended (but not required) resource, whether in print or electronic form. Links to a video series by the *Big Java* author, ostensibly covering the same material, were made available to students throughout the course. Additional materials such as past or current lecture notes or slides, past assignments and quizzes, review materials, etc. were also made available.

The other three sections were delivered using a more typical but modern approach blending lecture, live demos, in-class labs, and the interactive textbook *Programming with Java: Early Objects* [12], which we refer to throughout the remainder of this paper as a *traditional* approach. A third instructor (i.e., different from the Scrumage instructors) taught all traditional sections.

2.2 Instruments

Learning Attitudes. In consultation with the institution's teaching excellence center, we determined changes in student learning attitudes by adapting the "Motivated Strategies for Learning Questionnaire" [14], selecting 24 of 81 questions, to examine four categories: Effort Regulation, Metacognitive Self-Regulation, Help Seeking, and Control of Learning Beliefs. We also added questions specific to the types of learning materials typically available to students, e.g., "Watching online videos is an effective approach for me when I am learning new material." We deployed this instrument as an online survey in the second week of the course (once course registration had closed at the institution) and then again in the final week of the course. Students were given a nominal amount of extra credit for completing the survey, but the survey was an optional part of the coursework.

Content Acquisition. To determine student knowledge or content acquisition, we wrote a 12-question survey based on the *SCS1* assessment [13]. Although the original, validated instrument is intended to be independent of any specific programming language, many questions are written using a language similar to Python [24]. We found in pilots that students otherwise proficient in CS1 topics had some difficulty understanding some of the questions. We further wished to use a shorter instrument to more strongly motivate students to complete the survey (the original had 27 questions, often containing multiple parts comprising a single response). Like the learning attitudes survey, we deployed this knowledge acquisition survey in week 2 and in the final week of the course.

Retrospectives. Finally, students in the Scrumage sections completed *sprint retrospectives* at the conclusion of each sprint. The retrospectives invite students to reflect upon their experiences during the sprint and facilitate consideration of learning approaches for subsequent sprints. There is no natural analog to this instrument in the traditional course sections, so we use the responses to these retrospectives in our later analysis to further discuss the Scrumage section students' experience.

3 RESULTS

We report a variety of quantitative analyses applied to student results on the concepts and learning attitudes instruments. We also describe qualitative analysis of student comments collected on five retrospective surveys, administered at the end of each sprint.

3.1 Computer Science Concepts Test

Overall, 42 students completed both the start and end concepts instruments, with 43% of these students enrolled in Scrumage sections. The subsequent analysis focuses on these examples where pre- and post-assessments are available. Students in Scrumage sections outperformed traditional-section students by 15% on the post-test. Figure 1 shows how the distribution of scores changed from start to end using a kernel density estimate (KDE) for Scrumage (left) and Traditional (right) sections, with the solid curves showing starting scores and the dashed curves showing ending scores. In both pedagogies, improvement in content learning is statistically significant according to a Wilcoxon signed rank test (Scrumage p = .001 and Traditional p = .057). Among Scrumage students, improvement was greater on average and more uniformly distributed than among traditional students. In other words, lower performing students saw greater improvement in Scrumage than traditional classrooms. This effect can be seen in Figure 2, which shows how concepts test improvement was distributed among students with respect to their final grade in the course. For Scrumage students (left), content learning improvements are distributed more uniformly, while in Traditional sections (right) the content gains tend to be concentrated among the high-achieving students, with nearly 80% of the students who improved their score receiving an A in the course.

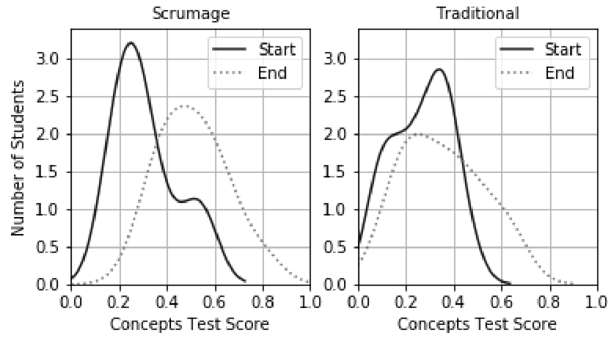

Figure 1: The distribution of scores using a Kernel Density Estimate from pre- and post-test (solid and dotted, respectively). Content learning is significant for both Scrumage (left, p=.001) and Traditional (right, p=.057).

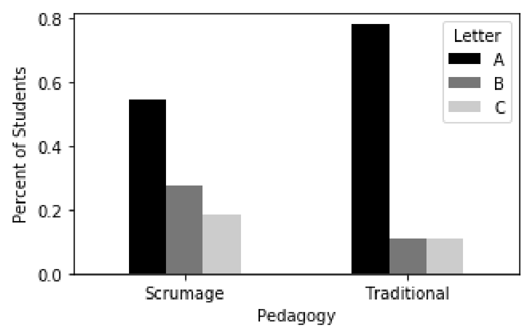

Figure 2: For each pedagogy, the distribution of students improving at least 10% on the concepts test is shown with respect to student final course grade.

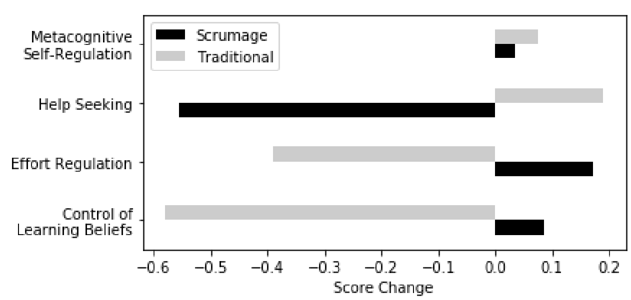

Figure 3: Within each Attitudes Survey category, the mean score change from pre- to post-test is shown for Scrumage (black) and Traditional (gray) students.

3.2 Learning Attitudes Survey

In total, 57 students completed both the start and end attitudes surveys. Of these, 56% of students were part of a Scrumage section of the course, with the remainder in traditional sections.

Comparing how student scores changed on average across each of the four main categories, there were several differences evident between the Scrumage and traditional sections of the course. Figure 3 shows the average change in score from pre- to post-test for each attitude category, and Figure 4 shows a KDE distribution of the changes in student scores over the course of the semester, with Scrumage in black and traditional in gray. Categorical changes were not statistically significant, except in two cases: Help Seeking in Scrumage students (p = .01) and the Control of Learning Beliefs for Traditional students (p = .05), which each decreased.

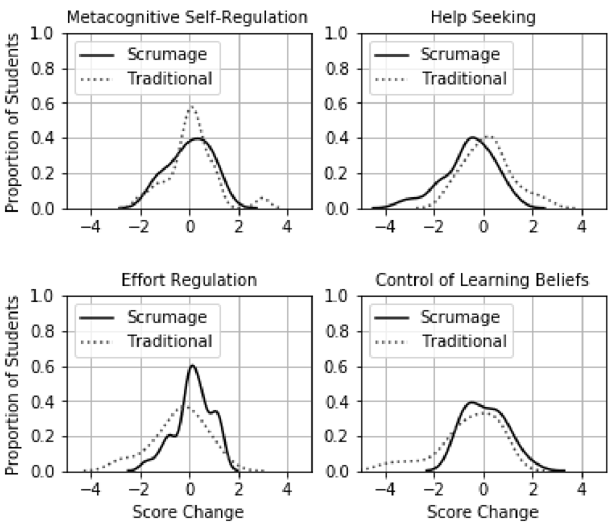

Figure 4: For each Attitudes Survey category, the distribution of score changes from pre- to post-test is shown for Scrumage (black) and Traditional (dotted) students using a Kernel Density Estimate.

We hypothesize that Scrumage students may have been less inclined to seek help due to the focus in Scrumage of providing students with a variety of resources in each sprint, empowering them to find answers on their own. Also, because students typically worked in teams, which offer a built-in mechanism for getting help, they may not have viewed working within their team as "seeking help." For questions in the Effort Regulation category (relating to students' perseverance and work ethic) and Control of Learning Beliefs (addressing students' feelings of being capable of learning the course material), Scrumage scores mildly improved on average, while traditional scores dropped more precipitously. It appears that many students in traditional computer science courses come to doubt their own ability to learn the material, irrespective of the amount of effort expended, while Scrumage students did not encounter the same frustrations. We conjecture that Scrumage students feel a greater sense of agency in their own learning. These results lead us to believe that using the Scrumage framework may help beginner programmers develop resilience and persistence in problem solving rather than adopting a negative attitude.

3.3 Jump Analysis

To get more insight into changes in students' learning attitudes, we counted "jumps," or large changes in student responses in individual survey questions. In the following analysis, we consider a student's response to have *improved* if the difference in score from pre- to post-survey was at least 2 (on a 1 to 7 Likert scale), to have *worsened* if the score dropped by at least 2, and to have stayed the *same* otherwise (a change between -1 and +1). Three questions in the Control of Learning category showed high numbers of students with these jumps between pre- and post-test (Table 1). Traditional students experienced a decline in their perception of control over the learning process that was not evident among Scrumage students. Likewise, three questions from the self-regulatory categories also evinced a large number of pre- to post-test jumps and similarly mirror the average trends visible in their respective categories (Table 2).

Table 1: Attitude Changes from Pre- to Post-test: Control of Learning

If I study in appropriate ways, then I will be able to learn the material in this course		*Improve*	*Same*	*Worse*
	Scrum.	8%	**89%**	**3%**
	Trad.	4%	**63%**	**33%**
It is my own fault if I don't learn the material in this course		*Improve*	*Same*	*Worse*
	Scrum.	9%	**86%**	**6%**
	Trad.	11%	**67%**	**22%**
If I try hard enough, then I will understand the course material		*Improve*	*Same*	*Worse*
	Scrum.	11%	**83%**	**6%**
	Trad.	11%	**63%**	**26%**

Table 2: Attitude Changes from Pre- to Post-test: Effort Regulation & Metacognitive Self-Regulation

When course work is difficult, I give up or only study the easy parts		*Improve*	*Same*	*Worse*
	Scrum.	8%	**89%**	**3%**
	Trad.	8%	**59%**	**33%**
I try to identify students in this class whom I can ask for help if necessary		*Improve*	*Same*	*Worse*
	Scrum.	**9%**	63%	**29%**
	Trad.	**26%**	67%	**7%**
I often find that I have been reading for class but don't know what it was all about		*Improve*	*Same*	*Worse*
	Scrum.	26%	**69%**	**6%**
	Trad.	30%	**48%**	**22%**

3.4 Attitude-Concepts Correlations

To identify how changes in student attitudes may interact with their learning of course material, we computed the Pearson standard correlation coefficient of the changes between pre- and post-attitude survey scores with the changes on the Computer Science concepts test. We chose 0.5 as the threshold indicating at least a moderate correlation.

Table 3: Attitude Questions Correlating to Content Learning for Scrumage Students

Correlation	Attitudes Question
0.65	When studying for this course I try to determine which concepts I don't understand well.
0.60	When reading for this course, I make up questions to help focus my reading.
0.57	Watching online videos is an effective approach for me when I am learning new material.
0.57	I ask the instructor to clarify concepts I don't understand well.
0.54	If course materials are difficult to understand, I change the way I read the material.
0.54	I try to change the way I study in order to fit the course requirements and instructor's teaching style.

Table 4: Coding Tags Applied to Student Retrospectives

Tag	*Description*	*Example Student Comment*
Achieve (+)	Student describes success at accomplishing a task.	"I learned how to use loops in a more advanced way."
Improve (+)	Student expresses feeling of improvement.	"I think I got better at evaluating problems"
Affirm (+)	Student affirms their own abilities.	"But once I fully understand..., I'm good at solving the problem."
Critical (-)	Student expresses doubt about their abilities.	"I also suck at coding."
Difficult (-)	Student comments on course being challenging.	"The math was hard."

For students experiencing the traditional pedagogy, only one question from the attitudes survey was correlated with a change in the concepts test score, with a correlation of +0.50: "Reading from a textbook is an effective approach for me when I am learning new material."

For Scrumage students, there were six questions above the moderate correlation threshold, as shown in Table 3. Interestingly, four of the six questions showing a positive correlation for the Scrumage students were from the Metacognitive Self-Regulation category. This finding offers some evidence that the Scrumage approach not only leads to a greater sense of learning control and responsibility, but that these positive attitudes then correlate positively with conceptual gains.

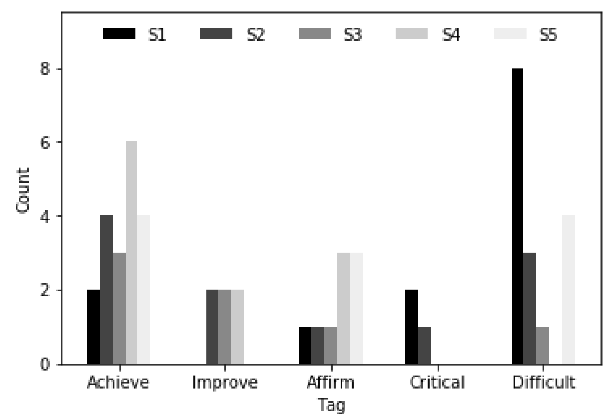

Figure 5: Counts of sentiment tags expressed in student retrospective comments for each sprint. Over time, incidence of positive tags increased while negative tags decreased.

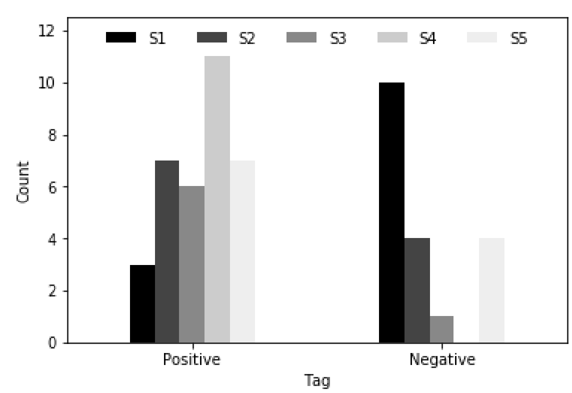

Figure 6: Counts of positive and negative student retrospective comments for each sprint.

3.5 Student Survey Comments

For the Scrumage students (though not the traditional students), we collected surveys following each sprint which included free text response fields for students to reflect on their learning experience in the course. The responses were hand-coded with five tags indicating commonly observed themes, either generally positive or generally negative in sentiment (Table 4). Incidence of positive tags (Achieve and Affirm), trended up over the semester while negative tags (Critical and Difficult) occurred less often in later sprints, except for the final sprint, (which shows an uptick in the Difficult comments). Figure 5 shows the frequency of observed tags over the course of the five sprints. The same trend is evident in Figure 6, which shows the overall positive or negative comments in aggregate over time. Even though coursework difficulty tends to increase as the semester progresses, the comments show the student attitudes generally trend more positive. Again, we believe this may be due to the fact that the Scrumage framework promotes personal development, and as students gain more learning strategies, they feel more confident and willing to persevere.

3.6 Student Perceptions of Learning Strategies

More resilience in learning attitudes may be a byproduct of having multiple resources for learning and regular reflection on progress. Student comments confirm that the Scrumage framework encourages a better understanding of how effective different course materials were for them. The learning strategy that showed the largest increase in preference from start to end survey was "jumping straight into problem solving," with multiple students indicating that this strategy was effective in retrospective comments.

Throughout the course, student comments show they were thinking critically about which approach to using the materials and class time was most successful for themselves individually. Not surprisingly, there was wide variation in preferences, with some students championing lectures, others preferring to read the text first, and still others choosing online videos. The lack of consensus indicates that the variety of resources offered in a Scrumage course will make the course more attractive to more students. Further, students were able to determine more complex strategies, giving specific times when a *change* in approach was effective, such as the comment "I learned that the textbook is a good place to go if I encountered an error." These comments support our findings from the attitudes survey that Scrumage does lead to a greater sense of learning control and responsibility. Further, based on our correlation analysis, this student empowerment is directly tied to conceptual gains.

4 THREATS TO VALIDITY

It is difficult if not impossible to make any two course sections truly "the same" with respect to content coverage. In two traditional sections, students in one section might be more or less prone to asking questions, perhaps leading to extended coverage of one topic in one section that is barely discussed in another section. By its very nature, Scrumage courses allow students to spend time at *their* discretion, facilitating very different experiences even for two students in the same section. The amount of exposure to any particular topic will vary from student to student because of the way each student decides to use in-class and out-of-class time. The three instructors agreed to cover the same content and to administer quizzes and examinations on the same concepts, though not at the same intervals and without usage of the exact same questions (to prevent any attempts at cheating).

It is further difficult to determine an ideal way to make a direct examination of two different learning techniques regardless of the number of instructors involved. Consider two approaches for two sections of the same course: (1) a single instructor teaches one section with Scrumage and with a traditional approach; and (2) two different instructors each teach one section, one with Scrumage and one with a traditional approach. In the former approach, an argument could be made that the instructor invested more time, effort and attention to one of the courses and that any observed differences are due to the instructor simply favoring one technique over the other. In the latter approach, an argument could be made that any observed differences are in fact due to the instructor differences rather than technique differences. Of the two approaches, we argue that the latter approach is superior because each instructor is trying to teach the course as best as possible within the technique. Scheduling and resource conflicts prevented a fuller examination in which the instructors switched roles in a subsequent semester, but this remains a possibility for the future.

Finally, we note here that one of the Scrumage course sections was populated exclusively by first-year university students whereas the others enrolled students at all undergraduate levels.

5 CONCLUSION AND FUTURE WORK

Scrumage is a novel agile teaching and learning methodology which aims to mimic real-world work expectations and promotes student autonomy. We found improvements with Scrumage over a traditional approach in both content learning and attitudes about learning. Moreover, the Scrumage framework seems to benefit students of all abilities in the class. Additionally, survey responses suggest that, even as content increases in difficulty, student attitudes improve over time. In a Scrumage class, students are better able to take responsibility for their own learning, and this positively correlates with better knowledge acquisition.

While the professors teaching with Scrumage followed the original conception of the Scrumage technique as outlined in prior work [3], individual instructors' implementations varied slightly, e.g., offering a "team bonus" for quiz performance, limiting team sizes, or including a "ramp up" period for students to acclimate to the method. These variations did not appear to affect student performance or attitudes. We believe it is the overall methodology and values of Scrumage (namely student choices in learning, rapid feedback, and reflection) that contribute to its success, and the details can be adapted on a class-by-class basis.

In the future we plan to continue using Scrumage in our Computer Science courses. Based on a successful initial pilot of the technique in an Analysis of Algorithms course, we believe the technique may be even more profitable for upper-level classes than for the introductory ones. We also have seen informally that there is a positive impact on professor experience in the course as well, with more engaged students during lectures, fewer students in office hours, and more preparedness for class. We would like to study the impact on professor as well as student in the future.

ACKNOWLEDGMENTS

The authors wish to thank the Elon University Center for the Advancement of Teaching and Learning, whose Scholar program aided in this work with both expertise and funding.

REFERENCES

[1] Lecia Barker and J. M. Cohoon. 2009. Key practices for retaining undergraduates in computing. National Center for Women and Information Technology, www. ncwit. org/retainundergrads.

[2] Mary Burgan. 2006. In Defense of Lecturing. Change: The Magazine of Higher Learning 38, 6: 30–34.

[3] Shannon Duvall, Dugald Ralph Hutchings, and Robert C. Duvall. 2018. Scrumage: A Method for Incorporating Multiple, Simultaneous Pedagogical Styles in the Classroom. In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18), 928–933.

[4] S. Duvall, D. Hutchings, and M. Kleckner. 2017. Changing perceptions of discrete mathematics through scrum-based course management practices. Journal of Computing Sciences in. Retrieved from https://dl.acm.org/citation.cfm?id=3144672

[5] Joyce Ehrlinger and E. Ashley Shain. 2014. How accuracy in students' self perceptions relates to success in learning. Acknowledgments and Dedication: 142.

[6] Nicholas W. Gelbar, Isaac Smith, and Brian Reichow. 2014. Systematic review of articles describing experience and supports of individuals with autism enrolled in college and university programs. Journal of autism and developmental disorders 44, 10: 2593–2601.

[7] Sue Grace and Phil Gravestock. 2008. Inclusion and diversity: Meeting the needs of all students. Routledge.

[8] Shaun R. Harper, Shaun R. Harper, and Stephen John Quaye. 2008. Student Engagement in Higher Education: Theoretical Perspectives and Practical Approaches for Diverse Populations. Routledge.

[9] Cay S. Horstmann. 2008. Big Java. Wiley.

[10] Alison King. 1993. From sage on the stage to guide on the side. College teaching 41, 1: 30–35.

[11] Jillian Kinzie, Robert Gonyea, Rick Shoup, and George D. Kuh. 2008. Promoting persistence and success of underrepresented students: Lessons for teaching and learning. New Directions for Teaching and Learning 2108, 115: 22–38.

[12] Roman Lysecky Adrian Lizarraga. 2018. Programming With Java: Early Objects.

[13] Miranda C. Parker, Mark Guzdial, and Shelly Engleman. 2016. Replication, Validation, and Use of a Language Independent CS1 Knowledge Assessment. In Proceedings of the 2016 ACM Conference on International Computing Education Research (ICER '16), 93–101.

[14] Paul R. Pintrich, Teresa Garcia, Wilbert James McKeachie, and David A. F. Smith. 1991. Motivated strategies for learning questionnaire. Regents of the University of Michigan.

[15] David Rose. 2000. Universal Design for Learning. Journal of Special Education Technology 15, 3: 45–49.

[16] David H. Rose, Wendy S. Harbour, Catherine Sam Johnston, Samantha G. Daley, and Linda Abarbanell. 2006. Universal design for learning in postsecondary education: Reflections on principles and their application. Journal of postsecondary education and disability 19, 2: 135–151.

[17] Viji Sathy and Kelly A. Hogan. 2019. How to Make Your Teaching More Inclusive. The Chronicle of Higher Education. Retrieved August 28, 2020 from https://www.chronicle.com/article/how-to-make-your-teaching-more-inclusive/

[18] Niral Shah, Colleen M. Lewis, Roxane Caires, Nasar Khan, Amirah Qureshi, Danielle Ehsanipour, and Noopur Gupta. 2013. Building Equitable Computer Science Classrooms: Elements of a Teaching Approach. In Proceeding of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE '13), 263–268.

[19] Scott Spurlock and Shannon Duvall. 2020. AlgoScrum: Improving an Algorithms Course with Scrumage. Journal of Computing Sciences in Colleges. To appear.

[20] Susan A. Ambrose and Marsha C. Lovett. 2014. Prior Knowledge is More Than Content: Skills and Beliefs Also Impact Learning. Applying Science of Learning in Education: 7–19.

[21] Jeff Sutherland and J. J. Sutherland. 2014. Scrum: The Art of Doing Twice the Work in Half the Time. Crown Publishing Group.

[22] Lynda Thomas, Mark Ratcliffe, John Woodbury, and Emma Jarman. 2002. Learning styles and performance in the introductory programming sequence. SIGCSE Bull. 34, 1: 33–37.

[23] Mary L. Wilson. 2012. Learning styles, instructional strategies, and the question of matching: A literature review. International Journal of Education 4, 3: 67.

[24] Welcome to Python.org. Python.org. Retrieved August 28, 2019 from http://python.org