Teaching Tip
Play Ball: Bringing Scrum into the Classroom

Jeffrey May
Computer Information Systems & Business Analytics
James Madison University
Harrisonburg, VA 22807, USA
mayjl@jmu.edu

Jim York
FoxHedge Ltd
18899 Maplewood Lane
Leesburg, VA 20175, USA
Jim@FoxHedgeLtd.com

Diane Lending
Computer Information Systems & Business Analytics
James Madison University
Harrisonburg, VA 22807, USA
lendindc@jmu.edu

ABSTRACT

Scrum has become a widely-used framework for technology development in both private industry and the government. As a result, Information Systems recruiters and executives have recently been placing a focus on students with Scrum knowledge. Unfortunately, current System Analysis and Design textbooks provide cursory attention to Scrum. Thus, the purpose of this paper is to suggest a starting point for teaching Scrum at the university level by presenting a classroom exercise (Ball Game) that can be used as a means for learning Scrum in more detail. This tip accomplishes three things: (1) introduces students to Scrum concepts with an engaging and memorable exercise, (2) provides a means for teaching students about estimation, and (3) offers an approach that allows students to witness firsthand how self-organized teams inspect, adapt, and evolve.

Keywords: Scrum, Systems analysis and design, Agile, Project management, Adaptive learning, Collaboration

1. INTRODUCTION

Scrum is an iterative and incremental framework containing simple roles, activities, artifacts, and rules founded on empirical process control theory. It has become a widely accepted agile framework in industry. For example, Capital One started an initiative in 2013 that called for a move to 80% Scrum, 20% Waterfall. The "10th Annual Survey of Agile Development" (Version One, 2016) found that 82% of respondents used Scrum or a Scrum variant in their organization.

As a result, Information Systems recruiters and executives have recently been placing a focus on students with Scrum knowledge. For example, Erica McDowell, a Booz Allen Hamilton executive states:

> In the last three years of my career I have yet to see one government RFP that did not include some form of a Scrum reference. These days, the Scrum framework and agile thinking have become the norm. Therefore, we place a strong emphasis on students who have been exposed to agile thinking in general and the Scrum framework in particular. (personal communications, April 22, 2015).

Unfortunately, current System Analysis and Design textbooks provide cursory attention to Scrum. Thus, the purpose of this paper is to introduce one way of teaching Scrum at the university level by presenting a classroom exercise (Ball Game) that can be used as a means for revealing various aspects of the Scrum framework.

Additionally, this paper will discuss the pedagogical value that this exercise offers both students and faculty. Ultimately, we view this paper as a starting point for a larger Scrum pedagogical research agenda.

The next section of this paper includes a brief background of the Scrum framework and ways that it has been used in university settings. We then introduce our exercise, the Ball Game, and provide suggestions on how this exercise can be incorporated into a Systems Analysis and Design course. Finally, we provide student, faculty, and recruiter reaction to the Ball Game in particular and the impacts of teaching Scrum in general.

2. SCRUM BACKGROUND

The Scrum framework originates from development processes created in Japan to enhance development speed and to provide flexibility for handling change (Takeuchi and Nonaka, 1986). Scrum was introduced in the United States by Ken Schwaber and Jeff Sutherland in 1995 at the annual OOPSLA (Object-Oriented Programming, Systems, Languages and Applications) conference (Sutherland et al., 2012). Schwaber and Sutherland (2016) define Scrum as a framework within which people can address complex, adaptive problems while productively and creatively delivering products of the highest possible value.

Scrum is one of many methodologies and frameworks that fall under the agile philosophy. The 4 basic tenets of agile place: (1) individuals and interactions over processes and tools, (2) working software over comprehensive documentation, (3) customer collaboration over contract negotiation, and (4) responding to change over following a plan (Beck et al., 2001). Additionally, Scrum is grounded by empirical process control theory as opposed to the defined process control model used with traditional approaches. In simplest terms, empirical process control theory posits that rich knowledge comes from what we learn through experience and places less focus on a priori assumptions or fixed plans. The notions of transparency, inspection, and adaptation are all common to agile thinking in general and the Scrum framework in particular (Vinekar and Huntley, 2010).

2.1 The Scrum Framework

Scrum is a technology development framework containing simple roles, activities, and artifacts. The three roles, as referenced in Figure 1, are the Product Owner (single cube), Scrum Master (whistle), and the Development Team (three cubes). As shown in Figure 1, the activities include Sprint Planning, Sprint Execution, Daily Scrum, Sprint Review, and a Sprint Retrospective. Finally, the artifacts include a Product Backlog, Sprint Backlog, and a Potentially Shippable Product Increment. It should be noted that organizations incorporate many other activities into the Scrum framework but the roles, activities, and artifacts shown in Figure 1 must be present and properly followed if a team wants to claim they are using Scrum. The remainder of Section 2.1 will briefly explain how all of the pieces shown in Figure 1 work together (see Schwaber and Sutherland (2016) for a more detailed explanation).

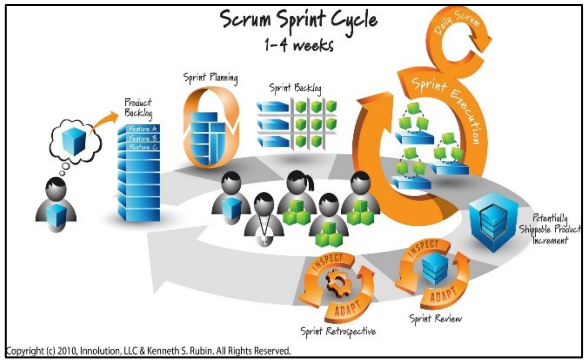


Figure 1: The Scrum Framework

In terms of **roles**, the Product Owner's primary responsibility is to maintain the integrity of the Product Backlog. Unlike a traditional project manager whose focus is on maintaining a balance between functionality, time, and cost, the Product Owner's primary focus is to ensure that the items (traditionally referred to as requirements) shown on the Product Backlog represent what the customer currently needs. The Scrum Master's primary responsibility is to remove any impediments that may be interfering with the Scrum Team. Unlike a project manager, the Scrum Master does not manage the Development Team; rather, she ensures that the team has what it needs to get work done. The Development Team is then responsible for the work needed to transform Product Backlog items into working features. They are a self-managing group of individuals with multiple talents that align with the general scope of a given project.

In terms of **artifacts**, the Product Backlog is a dynamic and ordered list of candidate work items that must be maintained throughout an entire product lifecycle. Unlike a traditional set of requirements, a Product Backlog constantly changes as the needs for the project become clearer to the customer. The Sprint Backlog is a smaller list of the highest value items from the Product Backlog that the Development Team has agreed to finish in the current sprint. As will be shown in Section 3, several iterations of work are required before the Development Team is capable of truly understanding the amount of work they can finish during a given interval. The Potentially Shippable Product Increment is a tangible chunk of work that has been completed from the customer's perspective. The Potentially Shippable Increment is available for inspection and feedback and could optionally be deployed offering immediate value to the customer.

In terms of **activities**, Sprint Planning is a meeting where the Development Team, Scrum Master, and Product Owner collaborate to decide what candidate Product Backlog Items the team will take on in the current Sprint. A Sprint is typically a 2-4 week period of time where the team attempts to finish the Sprint Backlog and create a Potentially Shippable Product. A Daily Scrum is held at the beginning of each work day for a maximum of 15 minutes where each team member discusses with the rest of the team what they got done the day before, what they will get done over the next day, and any impediments that are hindering them from getting work done. A Sprint Review is then held at the end of the Sprint and provides an opportunity for the customer to see, experience, and provide feedback on the Product

Increment. The feedback typically results in changes to improve the Product Backlog. These and all other changes to improve and maintain the Product Backlog fall into the general activity of Product Backlog Refinement (not shown in Figure 1), which can occur at any time. Finally, a Sprint Retrospective is a meeting where the team discusses what worked well and what might have worked better. The intent of the Sprint Retrospective is to identify and quickly implement incremental improvements to the team's process.

2.2 Pedagogical Value

Because of its success in industry, faculty (Baird and Riggins, 2012; Jiménez and Cliburn, 2016; Kropp, Meier, and Biddle, 2016; Pope-Ruark, 2012; Pope-Ruark et al., 2016; Wagh, 2012; Yue et al., 2009) are increasingly using the Scrum framework in the classroom to enhance project output and to stimulate rich collaborative environments. For example, Pope-Ruark et al. (2016) used Scrum for various university English class projects and stated, "Scrum could and in many cases should be used in any college course requiring collaboration, group projects, or problem solving." Within the IS field, Baird and Riggins (2012) used the Scrum framework for their capstone course project. Baird and Riggins (2012) found Scrum useful for maintaining student motivation due to more client interaction that forces accountability. Additionally, several European universities are modeling entire classes around a new teaching approach known as eduScrum, a framework that provides the foundation for teamwork throughout an entire class or semester (Delhij, van Solingen, and Wijnands, 2015).

Part of what makes Scrum promising for university settings is that it relies on an empowered, self-organizing team to discover, implement, and evolve the best process that works for them to accomplish a shared goal. In essence, a successful Scrum team acts as a complex, adaptive system changing from state to state (Blum and Li, 2008). Successful practice of the Scrum framework often times lead to holistic solutions but can only result from rich collaborative efforts that accept change as the norm rather than a hindrance. Thus, the highly valued skills of adaptation, problem-solving, and collaboration (Highsmith, 2013) can all be enriched if students successfully implement Scrum.

3. THE BALL GAME

In our Systems Analysis and Design class, after the students study and discuss the Scrum framework, we introduce a class exercise known as the Ball Game. The primary purpose of this exercise is for the students to experience for themselves the effects of a self-organizing team. Direct experience of the effects provides an opportunity to drive home the various elements of the Scrum framework and how it differs from traditional approaches.

3.1 The Setting

Our university is a public, medium-sized university in the mid-Atlantic region of the United States. Additionally, our Computer Information Systems department is within the School of Business. The Systems Analysis and Design (SAD) Course is taught primarily to seniors who are either Computer Information Systems majors or minors. We currently use an object-oriented focus to systems analysis and design and require the Dennis, Wixom, and Tegarden (2015) textbook as a reference.

We currently devote an entire week to Scrum at the beginning of the semester along with introducing the Ball Game and then proceed to teach traditional approaches and UML analysis modeling techniques. Throughout the semester, we compare and contrast what we are doing to what we would do if we were practicing Scrum. Finally, many of the assignments, given both in and out of the classroom, are done by students working primarily in groups. Two major class projects are assigned each semester and mimic the types of projects encountered in the consulting world. Currently, we do not require that the students practice the Scrum framework when working on team projects.

3.2 Rules of the Ball Game

Figure 2 illustrates the ball game in action with actual students. To begin the exercise, the class is divided into teams of six to twelve students. Each team is then given a bag that contains 20-30 hand-sized balls. The teams are then told that their goal is to deliver a maximum number of balls within a 2-minute period (Potentially Shippable Product). In order for a ball to count, it must be touched by every team member; it must spend some time airborne between touches; and when passed, it must not be passed between team members who are next to each other. If all of the balls have been used within the 2 minute time period (Sprint), the team may recycle balls and add to their total. The team is also responsible for an accurate count of balls actually delivered. At the conclusion of this explanation, the team is given two minutes to create their process and provide an estimate of how many balls they think they can deliver in a two-minute work timebox (Sprint Planning). When the two minute planning timebox expires, the work timebox begins. At the end of the work timebox, the team provides a count of their processed balls. The two-minute process creating and refining/two minute working cycle continues for five iterations.

Figure 2: The Ball Game in Action

3.3 Results

Figure 3 illustrates a typical team outcome after five trials. As shown in Figure 3, the estimation of the first trial is typically far from the actual results. This fact alone provides a great discussion point that could go in many directions. For example, one could discuss how this is similar to the

planning phase of the SDLC whereby original estimates are always wrong. Unfortunately, traditional approaches may place too much credence on initial estimates. In Scrum, the wrongness of initial estimates are factored into the development process along with dedicated periods to improve both process and estimation accuracy.

Figure 3 illustrates another key element of Scrum in particular and agile in general. That is, the notion of self-organizing teams. During Sprint Planning, Scrum Development Teams estimate how many items from the Product Backlog they should place on the Sprint Backlog. Ultimately, this means that they are agreeing to finish these items over a particular Sprint, yet it is very common for new Scrum Development Teams to provide inaccurate estimates during initial Sprints. However, as shown in Figure 3, as teams continue to work with each other and have a dedicated period (Sprint Retrospective) to discuss team enhancements, their ability to accurately estimate their work output quickly improves.

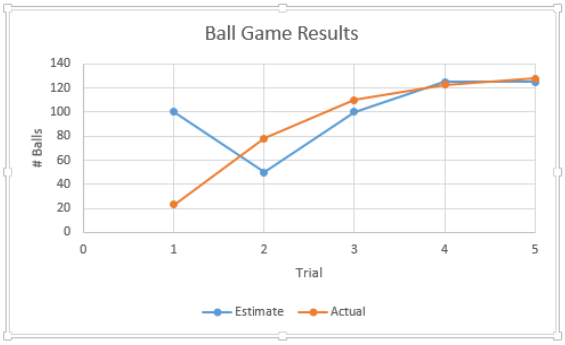


Figure 3: Ball Game Results

3.4 Teaching Suggestions and Further Discussion

Our first recommendation when using this teaching tip is to thoroughly ground the Ball Game with the roles, activities, and artifacts of the Scrum framework shown in Figure 1. We originally guide the students into learning Scrum for themselves through various reading exercises. After the Ball Game, we then lead them into making connections. In particular, we might ask the students to identify whether or not the various elements of Scrum can be related. For example, one might comment that the number of counted balls relates to a Potentially Shippable Product. The initial estimation period relates to Spring Planning. The subsequent discussion periods between trials relates to both Sprint Planning and a Sprint Retrospective. Or, each trial is a Sprint. Remain open and agile with the intention of bringing meaning to the various pieces of the Scrum framework.

Our second recommendation is to thoroughly discuss the concept of estimation. We drive home the point that initial estimations are always off. Thus, a little effort upfront helps a lot but a lot of effort upfront helps little. Because we still believe there is value teaching traditional techniques, you might consider placing this discussion in the context of the planning phase of the SDLC as well.

Our third recommendation is to use this exercise as a means for discussing various theoretical underpinnings of the agile philosophy. For example, you might discuss how the process improvements occurred over time because the teams began to understand each other. Of course, a Sprint Retrospective was required to communicate this understanding via individual inspection, but one might argue that a team eventually relies less on formal communication (i.e., Sprint Retrospective) over time as they are more capable of inspecting and adapting as a group (swarm intelligence). You might then consider discussing the differences between empirical process control and defined process control asking for advantages and disadvantages of each.

Our fourth recommendation is to use this exercise and the knowledge gained about the Scrum framework throughout the semester when discussing various traditional techniques. One holistic question might ask the students to identify if planning, analysis, design, and implementation are present in the Scrum framework. Hopefully the students will say that Scrum is heavy on planning and implementing but analysis and design are still present. Drive home the point that being agile and practicing Scrum does not lead to chaos; rather, agile frameworks are in fact bounded by rules. The major difference is the degree of this binding and one's ability to welcome change that is inevitable in today's fast paced environment.

Finally, we recommend that instructors who are serious about effectively teaching Scrum earn their Scrum Master's certification. Most of us have only read about Scrum and have not practiced it. Since part of a Scrum Master's role is to provide a development team with Scrum guidance, formal training helps you understand the foundations of Scrum, understand the reasons that Scrum works in industry, and gives you good stories to bring back to the classroom. The training class is two days and after you finish the training, you can take an online exam to get the certification.

4. STUDENT, FACULTY, AND RECRUITER REACTION

No doubt, our students' knowledge of the Scrum framework has improved drastically since adding a stronger focus on Scrum that transcends the cursory knowledge presented in our textbook. The Ball Game provides a fun and interactive means for learning Scrum and provides a memorable experience that can be used throughout the semester for comparisons and contrasts to various traditional and agile approaches to Systems Analysis and Design.

The reaction of our students to the Ball Game exercise in particular has been extremely positive. One student stated, "I liked how something so fun and challenging ended with a lesson that I will never forget….initial estimations are always wrong…so do your best but don't break your back!" Another student stated:

> It was really cool to see how the various teams progressed into well-oiled machines. I wish all of my past teams in college would have experienced the same success. We could never get past the negatives of our first encounters but then again we didn't have Sprint Retrospectives.

Another student stated:

> The Ball Game really got my adrenaline flowing, the professor then used it throughout the semester to make analogies to Scrum and the SDLC. I really believe it got me interested in Scrum which in turn has provided me an edge in interviews.

By understanding the value of Scrum and the competitive advantage it provides our students, three of our faculty members earned their Scrum Masters Certification and two went on to add a Product Owner Certification. Additionally, our Certified Scrum Trainer/Certified Enterprise Coach and coauthor, Jim York, volunteered to speak to our students and manage our first attempt at running the Ball Game. All of our SAD faculty agree that Scrum Master Certification for faculty is a necessary step to ensure maximizing the Scrum knowledge of our students. Additionally, we see a change in our own thinking as professors. For example, one of our professors was asked a typical question by a student, "What types of questions do you ask on tests?" His response was, "I have no plan that is set in stone, I'm agile, the questions will depend on where we go as a team."

Many of our recruiters have indicated that they have seen a notable change in our students' Scrum knowledge in general and their way of thinking in an agile manner. For example, one recruiter from KPMG mentioned, "In the past I used to ask question about Scrum and expected poor responses. However, now the students seem to talk about Scrum before I ask any questions. No doubt, this provides them an edge in my mind." Another recruiter from Booz Allen Hamilton commented, "It seems the students' think differently in interviews. Rather than stepping through answers with a rigid plan in mind, they seem more welcome to feedback and change. And they even tell me that they are agile!"

5. CONCLUSIONS

This paper explores the Scrum framework and provides a memorable exercise (Ball Game) for introducing Scrum into the Systems Analysis and Design Course. Along with providing a direct experience that illustrates the difficulties of estimating, the Ball Game provides a rich understanding of how self-organizing teams evolve and mature. Additionally, the Ball Game provides a memorable anchor that can be used throughout the semester when driving home Scrum knowledge and for comparing and contrasting Scrum with more traditional approaches.

We are convinced that Scrum in particular and agile in general are not merely fads, but should become a regular part of discussions in the Systems Analysis and Design course at all universities. Rather than simply read from a textbook, we believe that joining forces with industry and Scrum professionals enables the creation of experiential learning exercises that provide students with a competitive advantage.

6. REFERENCES

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.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile Software Development*.

Blum, C. & Li, X. (2008). Swarm Intelligence in Optimization. In Blum, C. and Merkle, D. (eds.). *Swarm Intelligence*, 43-85. Springer Berlin Heidelberg.

Delhij, A., van Solingen, R., & Wijnands, W. (2015). The eduScrum Guide, The Rules of the Game. Version 1.2 - September 2015, Reviewed by: Jeff Sutherland.

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

Highsmith, J. (2013). *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. Reading, MA: Addison-Wesley.

Jiménez, O. & Cliburn, D. (2016). Scrum in the Undergraduate Computer Science Curriculum. *Journal of Computing Sciences in Colleges*, 31(4), 108-114.

Kropp, M., Meier, A., & Biddle, R. (2016). Teaching Agile Collaboration Skills in the Classroom. In *2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*, 118-127.

Pope-Ruark, R. (2012). We Scrum Every Day: Using Scrum Project Management Framework for Group Projects. *College Teaching*, 60(4), 164-169.

Pope-Ruark, R., Eichel, M., Talbott, S., & Thornton, K. (2016). Let's Scrum: How Scrum Methodology Encourages Students to View Themselves as Collaborators. *Teaching and Learning Together in Higher Education*, 1(3).

Schwaber, K. & Sutherland, J. (2016). *The Definitive Guide to Scrum: The Rules of the Game*. Scrum.Org and ScrumInc.

Sutherland, J. V., Patel, D., Casanave, C., Miller, J., & Hollowell, G. (Eds.). (2012). Business Object Design and Implementation: OOPSLA'95 Workshop Proceedings, Austin, TX: Springer Science & Business Media.

Takeuchi, H. & Nonaka, I. (1986). The New Product Development Game. *Harvard Business Review*, 64(1), 137-146.

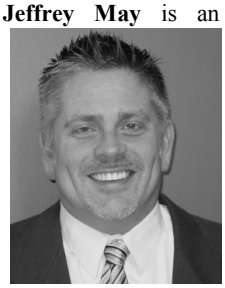
Version One. (2016). 10th Annual State of Agile Survey. Analysis Net Research.

Vinekar, V. & Huntley, C. (2010). Agility Versus Maturity: Is There Really a Trade-Off? *IEEE Computer*, 43(5).

Wagh, R. (2012). Using Scrum for Software Engineering Class Projects. In *2012 Agile India*, 68-71.

Yue, K. B., De Silva, D., Kim, D., Aktepe, M., Nagle, S., Boerger, C., & Verma, S. (2009). Building Real World Domain-Specific Social Network Websites as a Capstone Project. *Journal of Information Systems Education*, 20(1), 67-76.

AUTHOR BIOGRAPHIES



Jeffrey May is an Assistant Professor of Computer Information Systems and Business Analytics in the College of Business at James Madison University. His doctorate is in Information Systems from Virginia Commonwealth University. His research interests include multi-objective decision analysis techniques, socio-organizational IS security, and Information Systems pedagogy. Dr. May's research has been published in several journals including *Decision Support Systems and Electronic Commerce,* the *Journal of Information Systems Education*, and *the Journal of Computer Information Systems*. He currently is a certified Scrum Master and Product Owner.

For more than 25 years, **Jim York** has led, coached, and trained individuals, teams, and organizations in the implementation of Lean and Agile ideas. He has been the keynote and speaker at various Agile conferences and has held a number of leadership roles in the Agile community including the Scrum Alliance and the Agile Alliance. Jim is both a Certified Enterprise Coach℠ and a Certified Scrum Trainer®, one of only a handful of people world-wide to hold both of these top-level guide certifications from the Scrum Alliance. In 2007, he cofounded FoxHedge Ltd with Melissa York. Their mission is to help people balance agility and focus in building effective teams and products.

Diane Lending is the Capital One Information Security Faculty Scholar and Professor of Computer Information Systems at James Madison University in Harrisonburg Virginia. Her doctorate is in Management Information Systems from the University of Minnesota. Her research interests are in information systems education and requirements elicitation. Dr. Lending has written papers published in several journals including *Computers, Informatics, Nursing*; *Data Base*; *the Journal of Computer Information Systems*; and the *Journal of Information Systems Education*. Prior to joining academia, she was a programmer, systems analyst, and manager of systems development projects.

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 1055-3096