

Teaching Tip
**Scrum Boot Camp: Introducing Students to Agile System
Development**

Dmytro Babik

Recommended Citation: Babik, D. (2022). Teaching Tip: Scrum Boot Camp: Introducing Students to Agile System Development. *Journal of Information Systems Education*, 33(3), 195-208.

Article Link: <https://jise.org/Volume33/n3/JISE2022v33n3pp195-208.html>

Initial Submission:	7 March 2021
Minor Revision:	10 June 2021
Accepted:	24 August 2021
Published:	15 September 2022

Full terms and conditions of access and use, archived papers, submission instructions, a search tool, and much more can be found on the JISE website: <https://jise.org>

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

Teaching Tip

Scrum Boot Camp: Introducing Students to Agile System Development

Dmytro Babik

Department of Computer Information Systems and Business Analytics

James Madison University

Harrisonburg, VA 22807, USA

babikdx@jmu.edu

ABSTRACT

Agile system development approaches, such as Scrum, have become popular with a wide spectrum of organizations from start-ups to government agencies. Recruiters and executives have been seeking graduates with Scrum competency beyond cursory familiarity. CIS programs responded with a variety of activities and exercises to address this need. This teaching tip demonstrates a hands-on active-learning experience designed to advance students' understanding of Scrum values, roles, artifacts, and events; to build confidence of using it in practice; to expose students to the experience of using versatile software as a service (SaaS) tools for project management; to enable students' greater agency via self-forming teams and ability to control their workload; and to stimulate interest in Scrum and encourage students to seek professional certifications. The activity may be applied within a constrained time frame of a conventional systems analysis and design (SAD) or project management (PM) course and fits in the niche between a single-class activity and a semester project. Our evidence suggests that this learning experience achieves its pedagogical objectives, is effective in both face-to-face and synchronous online courses, and results in vastly positive student response.

Keywords: Systems analysis & design, Agile, Scrum, Team projects, Active learning, Collaborative learning

1. INTRODUCTION

Over the last decade, Agile system development practices, such as Scrum, Kanban, and Extreme Programming, have become popular among a wide spectrum of organizations, from startups to medium- and large-size enterprises, to government agencies and nonprofits (Petrova, 2019). Agile mindset and practices are attractive for timely and productive development of complex, competitive solutions in dynamic, ambiguous, and uncertain customer-driven environments. If executed competently and in a truly agile manner, Agile frameworks tend to outperform the traditional structured frameworks, such as the System Development Life Cycle (SDLC) and models, such as Waterfall (Terry, n.d.). Applicability of the Agile practices goes far beyond developing software products, reaching into fields such as marketing, HR, and finance (Edison et al., 2017; Johnson, 2014; Vandersluis, 2014). Consequently, academic programs in information systems, computer science, business, and other disciplines have responded to the industry's demands for graduates familiar with Agile practices (e.g., Scrum) by introducing Agile-related pedagogy to their curricula.

Although the popular Systems Analysis and Design (SAD) and Project Management (PM) textbooks still trail behind on including active learning materials for Scrum, there is now abundant literature on how to "do" Scrum, as well as how to train students in Scrum. This includes materials for practitioners

and learners (e.g., Eriksson, 2016), learning games (e.g., Kasperowski, 2014; PaulS, n.d.), and teaching practitioner publications. One of the challenges instructors and curriculum designers face is how to combine traditional and modern SAD and PM approaches in one course without crowding out schedules and materials. Another challenge is how to introduce difficult-to-teach, socially complex competencies needed for successful application of Agile philosophy to students without compromising Agile values. During 2015-2020, JISE published 13 papers on Agile/Scrum system development and project management, but only two teaching tips were published on how to introduce Scrum in the classroom setting – "Play Ball: Bringing Scrum into the Classroom" (May et al., 2016) and "Implementing Scrum Wholesale in the Classroom" (Baham, 2019). May et al. (2016) presented a single-class game activity (Ball Game) that introduces students to the notions of agility, adaptability, feedback, and estimation. Baham (2019) demonstrated how Scrum can be incorporated "wholesale" in a nine-week software development project in a SAD course. Both experiences aimed at improving students' *understanding* of Scrum, but the "wholesale" project also targeted building students' *confidence in using* Scrum in a real-world setting. Furthermore, Rush and Connolly (2020) proposed an interesting approach for teaching traditional PM-based content while using Scrum-style teamwork as the organizing logic for completing assignments in a semester PM course. They stressed

the need for learners to *internalize* Agile concepts through *active learning* in order to *gain confidence* in their use in practice. Active learning has been shown to improve student performance (Freeman et al., 2014), and various such exercises have long been used in project management courses (Allan, 1999).

As many SAD and PM courses attempt to introduce Agile-related content and balance it with traditional content, such as SDLC, UML, and the Unified Process, the challenge is to find exercises with the right focus and load. Larger projects tend to focus on the product rather than the process, while simpler and shorter activities tend to be too superficial to provide an in-depth understanding of Agile. Overall, an instructor who tries to incorporate active learning Scrum experiences beyond a cursory introduction or a quick gaming exercise without engaging in a semester project, may find a lack of such activities. To address this gap, the author of this paper developed a “lighter” Scrum active learning exercise (dubbed “Scrum Boot Camp Project”) that fits the niche between a “light” single-class activity and a “heavy” semester “wholesale” project. Several transformations in our undergraduate SAD course led to experimenting with Scrum-focused activities and pedagogies:

- transitioning from the SDLC-focused instruction to introducing a diverse arsenal of system development and project management frameworks;
- “flipping the classroom,” i.e., engaging students in active learning during in-class interactions with the instructor (and leaving passive learning activities, such as reading and completing quizzes for “home consumption”);
- seeking to grant students more agency in creative choice and catering to various majors and minors;
- encouraging students to consider training for professional Agile certifications.

The purpose of this teaching tip is to present a hands-on experience in Scrum that may be applied within a constrained time frame of a conventional SAD or PM course while advancing students’ understanding of Scrum beyond memorization of abstract material from a textbook. This teaching tip is addressed to the instructors already familiar with Agile and Scrum who seek to introduce students to Scrum in an undergraduate SAD or PM course through a three-week team project. The value of this teaching tip is in demonstrating how to: (a) facilitate solid understanding of Scrum values, accountabilities/roles, artifacts, and events; (b) expose students to the experience of using popular and versatile SaaS tools for project management (e.g., Trello); (c) allow students to be in control of choosing teams, projects, and workload; and (d) stimulate interest in Scrum and encourage student to seek professional certifications.

Section 2 brings attention to several misconceptions that may emerge when introducing Scrum in a college course. Section 3 describes the Scrum Boot Camp learning experience. Section 4 further elaborates on some pitfalls and lessons learned and provides recommendations to address them. Section 5 draws the conclusions.

2. WHAT SCRUM IS NOT

Agile philosophy and frameworks, such as Scrum, have been extensively described and discussed in the literature (Saltz & Heckman, 2020; Sharp et al., 2020; Topi & Spurrier, 2019).

However, one important issue is usually overlooked when introducing Agile and Scrum to students in a college course. Scrum has been widely referred to in the academic (and, at times, in the practitioner) literature as an Agile system development *method* or a *methodology*, similarly to how the Unified Process is often described as a methodology rather than a framework. Although a scrupulous description of the team accountabilities/roles, artifacts, and events by the Scrum Guide (Schwaber & Sutherland, 2020) simplifies understanding of this framework, it also leads many to believe that Scrum follows a *defined process* model of control (Bequette, 2002) in the sense that if set up and run methodically, it will produce a desired result for a given set of inputs (Finkel, 2016). Many practitioners and coaches caution that this is a misinterpretation and misapplication of Scrum that results in “Dark Scrum,” in which the processes and people are treated too “administratively,” leading to micromanagement, misplaced incentives, adverse behaviors, failed results, and vocal disappointment and resistance (González Sanchis, 2014; Jeffries, 2016; Margetak, 2019). Despite a well-outlined framework structure, Scrum is also often described as “chaotic” (Dennis et al., 2015). This description was used by Sutherland and Schwaber (2007) to emphasize Scrum’s *empirical process* model of control and applicability to volatile, uncertain, complex, and ambiguous (VUCA) conditions (Finkel, 2016; Mack et al., 2015; Stuart, 2020). Instead, “chaotic” is often interpreted as lacking coordination and management, with “programmers gone wild” (Dennis et al., 2015, p. 13). Lost in this view is that Scrum is meant to be a barebone set of principles and guidelines to minimize chaos in learning and discovery in a VUCA environment and to enable the empirical process model of control (in simple words, discover and use what works, and discard what does not).

Students who are indoctrinated into the “follow the rules” and “textbooks contain objective truth” mindset, often struggle to understand the essence of Scrum as an empirical process framework when trying to navigate these conflicting views. They tend to accept more easily the idea of “Agile methodology,” Scrum “ceremonies,” and completing tasks to do “what the instructor wants,” rather than the idea of a heuristic framework based on initiative, self-organization, self-management, and mutual respect to tackle uncertainty and to solve complex problems. As a result, when given an opportunity to practice Scrum, student teams may also descend into “Dark Scrum.” For example, in the upper-level elective project-based course “Agile Development of IoT Solutions,” the author observed student teams to perform Scrum “ceremonies” just to keep the instructors happy, while engaging in other team interactions to actually have things done and complete the project. Our goal is to help students develop a critical analysis mindset and the scientific method discipline to discover what works and what does not when applying the Scrum framework.

Industry evidence suggests that Scrum fails if it is either overly “chaotic” and the team members struggle to fulfill their roles for the benefits of the team’s success or if it is excessively structured, “managed,” and executed in the overly administrative manner, in which the roles and “rituals” are performed only because the team is “required” to use Scrum, rather than finds it to be the most natural way to operate (Baars, n.d.; Gwosdz, 2020; nTask, 2020; Wischweh, 2019). We designed and implemented the Scrum Boot Camp project in our

SAD course to provide students with a hands-on trial-and-error experience through which they could actively learn the values of Agile philosophy and how to avoid both extremes while applying the Scrum framework.

3. SCRUM BOOT CAMP PROJECT

3.1 The Context

Our upper-level undergraduate “Systems Analysis and Design” course historically has been focused on the traditional structured design frameworks, such as the SDLC, and models like Waterfall, with their distinct system development phases. The course includes a sequence of three team teaching-case-based projects emulating Waterfall phases – Planning, Analysis, Design, and Implementation. Usually, the course has several face-to-face sections taught by several instructors, with 20-30 students per section. By completing these projects, students gain hands-on experience of creating a Project Plan, gathering system requirements through document analysis and interviews, creating process, functional, and structural system models with the UML, and drafting a System Proposal. Each of these projects takes between two and four weeks, with the total duration of the sequence being about eight weeks. The improved learning outcomes of this deliberate and assessment-focused approach have been demonstrated by Lending et al. (2018).

Since 2016, we have been gradually re-orienting the course to include the introduction to Agile system development in a more substantive manner than just briefly covering the topic through the textbook and multiple-choice testing (several faculty members teaching this course also concluded Scrum training and obtained CSM certifications). The Ball Game is one example of an interactive and engaged-learning activity helping us in this transition (May et al., 2016). The Scrum Boot Camp Project described in this paper is another novel active-learning experience shown to be effective and popular with students. The primary objective of this 2-3 week exercise was two-fold: (a) to provide students with hands-on experience of Scrum application by self-organized and self-managing teams, and (b) cultivate students’ Agile mindset and understanding of Scrum as an empirical process control framework rather than a defined process methodology.

3.2 Implementation

3.2.1 Overview. During the introductory weeks of the course, students are exposed to a variety of system development philosophies (structured vs. agile), frameworks (SDLC, RAD, Scrum), and methodologies through required reading (such as Manifesto for Agile Software Development, 2001; Dennis et al., 2015; Schwaber & Sutherland, 2020), various video materials, in-class discussions, and an active learning exercise (Ball Game). They also complete several quizzes and a writing assignment for formative and summative assessment of their comprehension of the topic. This period is followed by a deeper dive into the SDLC through a sequence of team projects described above. Upon completion of these “Waterfall” projects (and typically about four weeks before the end of the semester), students are introduced to the Scrum Boot Camp project (see Appendix A for the sample instructions posted in Canvas).

The overall premise of this project is that self-organized and self-managing student teams build two increments of the

product of their own choice in two one-week sprints. The nature, complexity, and scope of the product are not dictated by the instructor, but students are provided some non-specific examples of possible projects (e.g., a simple computer game, a website, an IoT device, a digital cookbook, a gingerbread house, or an origami). Teams are not expected to produce a fully completed product, but rather a functioning product increment, with the possibility of further development. Students are encouraged to ask questions and consult with the instructor about the viability of their ideas before the beginning of the first sprint.

The point of this self-regulated approach is to allow students to remain in control of their workload during the busiest time of the semester (or even the busiest time of their college career for some students, as many of them are also simultaneously finalizing their capstone projects). The primary pedagogical objective of the project is not to have the team deliver products of certain level of complexity and scope, but rather to engage them in active ideation and teamwork, to illustrate the importance of Scrum values, accountabilities, artifacts, and events, and to convey the idea that in Agile development, the agency for selecting a project and self-managing lies with the teams.

3.2.2 Team Formation. Teams form over the course of several days between posting the project instructions and the beginning of Sprint 1. Unlike the three “Waterfall” projects, where teams are assigned by the instructor (to give a sense of the traditional administrative nature of the SDLC) and re-shuffled for each of the projects (to reduce dissatisfaction due to adverse behaviors, such as “free-riding,” and to allow “cross-pollination” of ideas), in the “Scrum” project, students form the teams as they please. The team size is not dictated, but students are informed that a team cannot be smaller than four people, and that it may be difficult to coordinate more than six people.

We have experimented with two approaches to self-forming teams. One approach was to use a variation of the Name Line game (“Name Line,” 2019): students and the instructor stand in a circle, and the instructor explains the rules; each participant, beginning with the instructor, has to say their name, one skill they have, and one thing they dream to build. They simultaneously make movements to illustrate the skill and the thing; the next person must say previous persons’ names, skills, and things, and introduce themselves. This activity repeats until the last person; then it continues so that the participants earlier in the circle must recall and introduce as many others as the participants later in the circle. During this usually very entertaining activity, students get to share their skills and interests and form preferences for organizing teams. Typically, students with attractive ideas emerge as Product Owners, whereas the most outgoing, sociable, and engaged students emerge as Scrum Masters. This activity, however, takes time, especially in a large class, and is only suitable for face-to-face classes. In the online classes, team formation relies on the fact that during the “Waterfall” projects, teams are re-shuffled so that each student gets to know as many classmates as possible and to form their preferences for the Scrum team. Once a team is formed, it needs to brainstorm and agree on the product idea, team name, roles, and pass this information on to the instructor.

3.2.3 Accountabilities/Roles. Every team includes a Product Owner (PO), a Scrum Master (SM), and several Developers. It

is worth noting that the 2020 version of the Scrum Guide avoids the use of the term “role” and replaces it with “accountability” (Schwaber & Sutherland, 2020). The PO’s responsibilities include explaining the product idea to the team, gathering additional ideas from team members, and forming the team’s consensus on building the product. The PO is in charge of setting up the Scrum Board in Trello, adding an initial set of items (ideas for increments, user stories, product requirements) to the Product Backlog, and constantly curating the Product Backlog (revising, refining, and reprioritizing backlog items). The PO also leads the Sprint Review in which the team demonstrates their Product Increment to the class, answers questions, and notes suggestions.

Developers conduct all activities of creating and testing Product Increments. The SM’s primary responsibility is to act as a *leader who serves the team* and to help the team maximize the value of its efforts by facilitating interactions, helping resolve any issues, and removing any obstacles the team faces (“*Servant leader*” in the Scrum Guide 2017; “*a leader who serves the team*” in the Scrum Guide 2020). In our experience, students unfamiliar with Scrum tend to misunderstand the accountabilities and look up to either the PO or the SM as a “Project Manager.” Therefore, they need to be constantly reminded that the role of the Project Manager (who tell others what to do) does not exist in Scrum, there is no “chain of command” or responsibility hierarchy within a team, and all team members should practice proactive thinking and behavior.

3.2.4 Artifacts. The Scrum Boot Camp project introduces students to the Scrum artifacts – *Product Backlog*, *Sprint Backlog*, and a (potentially releasable) *Product Increment*. Product Backlog and Sprint Backlog are facilitated in Trello (trello.com), a web-based, Kanban-style, list-making SaaS provided by Atlassian (Figure 1). Although the PO is responsible for setting up the Trello Scrum Board and Backlogs, all team members are encouraged to actively make appropriate changes in it (e.g., to add backlog item cards, to-do lists, deadlines, attachments, labels; to record activities, to move cards, etc.). Trello is a free, lightweight, easy-to-learn-and-use, and highly interactive tool that integrates well with other communication and task management tools (email, calendar, cloud storage, Slack, GitHub, etc.). In this project, students must build Product Increments that are then demonstrated during in-class Sprint Review.

3.2.5 Events. The Scrum Boot Camp activities are based on the Scrum events – *Sprint*, *Sprint Planning*, *Daily Scrum*, *Sprint Review*, and *Sprint Retrospective*. In the literature, Scrum events are often referred to as “rituals” or “ceremonies.” We do not like using these terms, because students may apply their traditional meanings – something you do, not because it is meaningful to you, but because you need to do it to satisfy somebody else – to understand the Scrum events. When this happens, the Scrum “rituals” are no longer useful nor make sense.

In our implementation of the Scrum Boot Camp, teams complete two one-week Sprints, each resulting in a demonstrable Product Increment. Thus, together with pre-project activities, the entire Boot Camp project takes about six class meetings (three weeks with two classes per week; see a sample schedule in Figure 2). Class meetings are structured so that teams can work on their Product Increments in and outside

of class. This provides the instructor with the opportunity to observe team interactions and answer questions, as well as allows students to interact for some time without being observed.

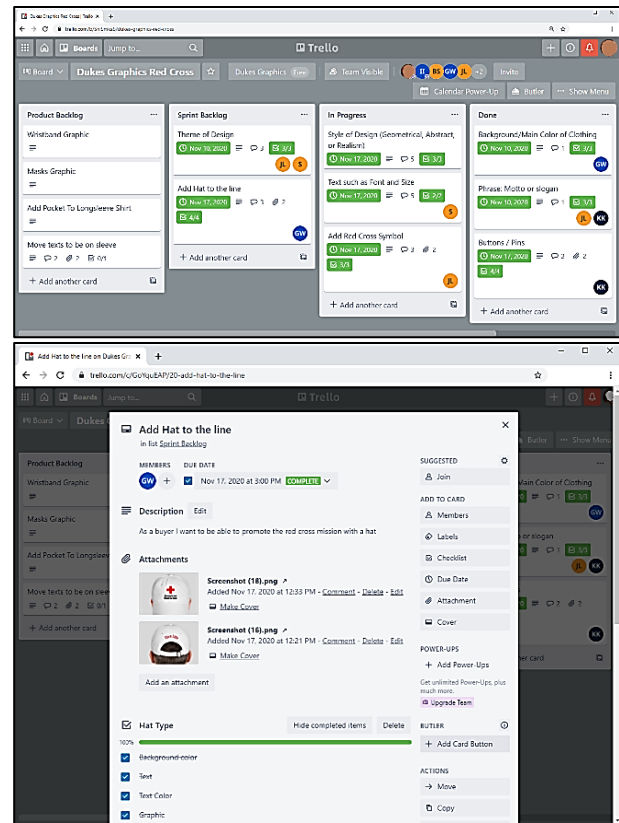


Figure 1. Sample Scrum Board and Backlog Item Card in Trello

Every Sprint begins with in-class Sprint Planning. Teams break out to review their Product Backlogs and to identify high-priority and most-well-defined items for their Product Increments. Although user story is a general Agile notion and is optional in Scrum, in our implementation, we train students to state and refine requirements and features as user stories that follow this template “*As a <user role>, I want to be able to <do what?> so that <why?>.*” Each user story is recorded as a backlog item in a Trello Board card. As user stories evolve, the cards may be split, combined, and appended with tasks, checklists, attachments, etc. Based on the team consensus, the selected Product Backlog items are moved to the Sprint Backlog. The PO must make a case for each item, and the Developers should agree that they can commit to implementing these items during the current Sprint. The SM facilitates this discussion, so that it is conducted in a professional, friendly, agile, and productive manner. In our experience, a 20-minute Sprint Planning event is sufficient to select three to six items for the current Sprint. The key checkpoints for Sprint Planning are: (a) the user stories must be present in the Product Backlog and be well articulated by the PO before the event; (b) upon the end of Sprint Planning, the selected user stories must be in the Sprint Backlog; (c) Sprint Planning must be “time-boxed,” i.e., once the duration of these events was determined (by the instructor

or through the negotiation of the instructor with the teams), Sprint Planning could not exceed the set duration; it is SM's responsibility to ensure that the team manages to complete their Sprint Planning on time. To help students understand the importance and expectations for Sprint Planning, Sprint 1 Planning is preceded by a brief discussion and Q&A. During Sprint Planning, the instructor observes, answers teams' questions, clarifies, or guides SMs to facilitate the event.

Time	Event	Activities
Week 1, Class meeting 1	Getting started	Pre-project activities: forming teams, determining product, discussing and agreeing on accountabilities, setting up Trello Board, initial brainstorming and populating Product Backlog
Week 2, Class meeting 2	Sprint 1 Planning	Discussing the purpose, result, and meaning of Sprint Planning (10 minutes). Sprint 1 Planning in teams (20 minutes)
Week 2, Class meeting 3	Daily Scrum	Daily Scrum (10 minutes), in-class work on the project, consultations with the instructor
Week 3, Class meeting 4	Sprint 1 Review Sprint 1 Retrospective Sprint 2 Planning	Sprint 1 Review (5 minutes per team), Sprint 1 Retrospective (10 minutes), Sprint 2 Planning (20 minutes)
Week 3, Class meeting 5	Daily Scrum	Daily Scrum (10 minutes), in-class work on the project, consultations with the instructor
Week 4, Class meeting 6	Sprint 2 Review Sprint 2 Retrospective Class Debrief	Sprint 2 Review (5 minutes per team), Sprint 2 Retrospective (10 minutes), Class Debrief (obtaining students' feedback, discussing lessons learned) Peer evaluations of team-member contributions (after-class activity)

Figure 2. Sample Scrum Boot Camp Schedule

Each Sprint closes with two events – Sprint Review and Sprint Retrospective. In our implementation of the Sprint Review, each team demonstrates their Product Increment to the

rest of the class in a highly interactive and “time-boxed” manner. The PO leads the demonstration, and the rest of the team should be actively involved; the team is encouraged to show the implemented user stories in the Trello Board and explain how they went about implementing them; the audience is encouraged to ask questions and make suggestions.

Based on the typical scope of the projects and to encourage fast-paced and dynamic discussion, we “time-box” the Sprint Review to five minutes per team. The key checkpoints for the Sprint Review are: (a) the Product Increment must have practical utility to the users, e.g., perform useful functions; students must understand that a Product Increment is not a “prototype,” sketch, or model, but a potentially deployable functioning version; (b) the user stories that were in the Sprint Backlog must be moved to the Done list by the time of Review; any incomplete Sprint Backlog items must be explained by the team; the instructor needs to stress the importance of accurate estimation and guide the team on how to deal with situations when a user story could not be completed by the end of the Sprint; (c) the user story cards should reflect the team's work (e.g., assigned responsibilities, activity notes and messages, marked checklists, attached digital artifacts, labels, deadlines); (d) any discussions of the issues unrelated to the Product Increment should be postponed until the Sprint Retrospective; (e) the Sprint Review must be “time-boxed.”

Immediately after the Sprint Review, teams break out for the Sprint Retrospective. These are “time-boxed” team-member-only conversations (in our case, 10-minutes) during which teams discuss the completed Sprint, identify any issues (e.g., negative team dynamics, insufficiently performed roles, scheduling and other conflicts, lacking resources), and develop strategies to resolve these issues. The team should also acknowledge their successes and discuss how to leverage them. The instructor observes and gently guides as necessary, but most of the facilitation should be done by the SM. The key checkpoints for the Sprint Retrospective are: (a) the conversation, facilitated by SM, should occur in a friendly and professional manner; (b) all team members should be actively involved; (c) the conversation should focus on the team process rather than on the product; any discussion of user stories and the completed or forthcoming Product Increments should be postponed until next Sprint Planning. Just as with Sprint 1 Planning, it is helpful to precede the Sprint 1 Review and Retrospective with a brief all-class discussion and Q&A of the purpose and expectations for these events.

Daily Scrums (or Standups) are probably the most problematic events to conduct in the context of the college course because students are not meeting daily and are not committed full-time to their projects. Most likely, students are simultaneously committed to multiple team projects in different courses (including capstones). In the industry context, during a Sprint, Daily Scrums are conducted every morning for 15 minutes. During these brief “standups,” the team gauges their progress towards the Sprint goal (Product Increment) and produces an actionable plan for the next day of work, usually, by each team member reporting three things: what they had done yesterday, what they will do today; what obstacles they face and what they will do to overcome them (Schwaber & Sutherland, 2020; SCRUMstudy, 2017) (Figure 3). In the context of a college course project, simulating Daily Scrums takes a bit of creativity. We considered and tried several options: (a) a one-per-sprint, in-class, face-to-face standup

where the team conducts this event in class and can be observed by the instructor; (b) a virtual daily standup (over phone, Skype, Zoom, etc.) in which the team meets daily but not face-to-face; (c) daily Trello board posts, in which each team member is expected to add an activity note in the user story cards documenting their daily activities.



Figure 3. Daily Scrum

Source: <https://medium.com/@thofhan/understanding-scrum-value-a9ac965c4dd9>, in public domain

Given the typical scope of the team's products and the simulated nature of the Scrum Boot Camp project, the first option (one-per-sprint, in-class, face-to-face standup) appears to work best. The advantage of this option is that teams are given an opportunity to practice Daily Scrum in the manner closest to how it would be conducted in the workplace; the instructor can observe team dynamics and guide it if necessary. The disadvantage is that the "Daily" Scrum becomes weekly, rather than daily; however, considering the factors described above, this is not detrimental to students' learning of the importance and mechanics of this event; the "Scrum fatigue" and overload are much more likely to adversely affect students' reception of this learning experience. Considering the typically minimal complexity and scope of the products, our in-class Daily Scrums are five-minutes (rather than 15-minutes); this provides enough time for every team member to report and for the instructor to observe the team dynamics. The key checkpoints for the Daily Scrums are: (a) the discussion, facilitated by the SM should occur in a dynamic, friendly, and professional manner; (b) all team members should be actively involved, and (c) the three questions should be answered by each team member.

3.2.6 Performance Assessment. The primary goal of assessment in the Scrum Boot Camp project is to evaluate students' understanding of the Scrum framework, the mechanics of implementing Scrum, and the values of Agile philosophy. The focus is on the team process rather than on the scope and quality of the product. We use two assessment tools: the rubric (Appendix C) and peer evaluations of team-member contributions (which also includes self-assessment and self-reflection). The rubric captures team performance (as a team score) based on the instructor's observations of accountabilities, artifacts, and events. Every team member receives that score as their base grade for the project. This grade may be individually adjusted up- or downwards at the

instructor's discretion based on the quantitative data and qualitative comments in peer evaluations. To conduct assessment efficiently, the rubric is used through a Learning Management System (LMS, Canvas in our case); peer evaluations are administered through an online peer review and assessment (OPRA) system (we use Mobius SLIP, mobiusSLIP.com, Babik et al., 2016, 2017); other OPRA systems could also be used, e.g., CATME, catme.org, Brutus & Donia, 2010). Using an OPRA system helps quickly identify and corroborate exceptional team leaders and contributors, as well as non-contributing or free-riding team members through learning analytics and written comments (see Figure 4).

3.3 Students' Feedback and Evidence of Efficacy

Since we introduced the Scrum Boot Camp project in our SAD course in spring 2019, we observed a vast and remarkably positive reception from students through self-reflections, project debriefs, and course evaluations. Importantly, we have never received any negative or disgruntled comments. Moreover, at the conclusion of the project, several students (in a typical class of 90) each semester inquire about Scrum certifications, such as Certified Scrum Master (CSM) and Certified Scrum Product Owner (CSPO), or relate the project to other experiences, such as working in a Scrum team during an internship or discussing Scrum during a job interview. We noted students' enthusiasm about this project, as many of them view it as an important credential for their job search and future career. Typically, each semester, several students indicate in their course evaluations that this project was their favorite part of the course.

We often observe some anxiety among students before their teams agree on the products and accountabilities, and begin ideating and creating their Product Backlogs. During Sprint 1, though, we usually observe emotional relief about the workload and heightened enthusiasm about building something tangible and demonstrable. We noted the following sentiments in students' reactions (quotes here are from students' self-reflections):

- Enjoyment of learning about Scrum and project management tools:
 - "[The project] helped me see Scrum in a 'good' light - it is no longer 'scary' and I believe I could confidently try using this in a workplace setting with a team." (Fall 2020)
 - "Overall, it was a great experience working with my Scrum team to build our Scrum project. [...] I have definitely gained hands-on experience of working under Scrum framework and have learned so much about it." (Fall 2019)
 - "I was very satisfied with the project and my team. [...] I liked the project because it did teach me a lot about the Scrum Framework. The most valuable thing I learned from this project is about Trello Boards. I was not familiar with them before, but my team tried to use as many of the features that there were to offer. Really great experience!!" (Fall 2020)

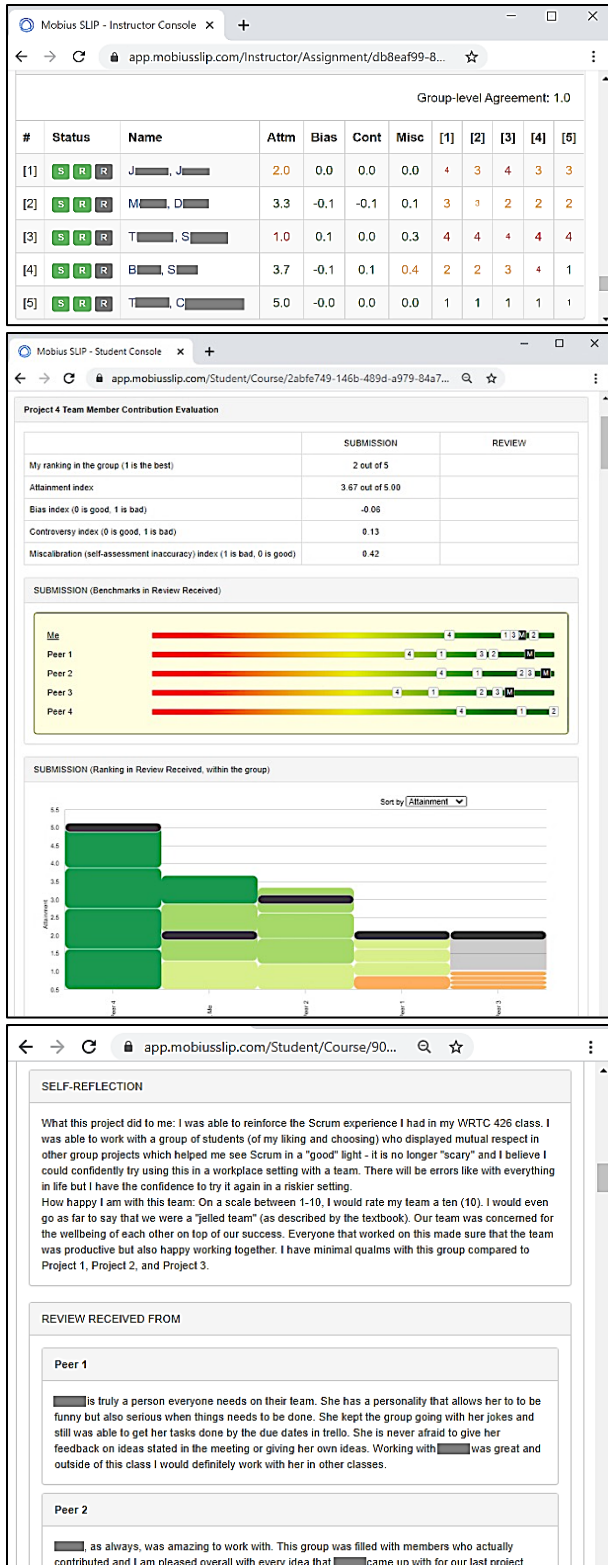


Figure 4. Sample Peer Evaluations of Team-Member Contributions in Mobius SLIP

- “From this project I got a feel for how scrum differs

from other systems development frameworks. I got to learn [...] a scrum process, an experience I enjoyed. In addition, I learned a new software, I thought Trello was an interesting software that holds a great ability for organization and collaboration between groups. [...] This project was a lot of fun and I am excited with the final product.” (Fall 2020)

- “[Scrum] was new to me. [...] I am very glad that we did this now because I feel like I have gotten my feet wet in regards to the process of using scrum. I know I am going to greatly appreciate having done this when I get into the workplace.” (Fall 2020)
- “I am very satisfied with the outcome of this project. Even though I had used Scrum before, getting more practice with it by completing a few sprints was very useful.” (Fall 2020)
- “[...] I now have deeper understanding for how Scrum works. Actually creating a product and going through 2 sprints taught me more about Scrum than watching videos on YouTube about it. I would've liked if we transitioned to Scrum from Waterfall earlier in the semester, so that we had half-Waterfall, half-Scrum. [...] I do think it is useful to us [...] going into the 'real' world, as many workplaces are implementing Scrum/Agile.” (Fall 2020)
- Enjoyment of working with the team and on the product of student’s own choice:
 - “This project was by far my favorite project in this course. We were able to have fun and learn at the same time. [...] I enjoyed everyone in my team. We all brought something different to the team and that's what I think made us work so well together.” (Fall 2020)
 - “I would [...] say that we were a 'jelled team'. Our team was concerned for the wellbeing of each other on top of our success. Everyone that worked on this made sure that the team was productive but also happy working together. I have minimal qualms with this [team].” (Fall 2020)
- Enjoyment of observing other teams creating different interesting products:
 - “Overall this was a great experience, and it was really cool to choose our own topic/product to explore scrum. It was interesting too to see what other teams were working on and their progress because [each] team had a different product.” (Fall 2020)
- Enjoyment of being in control of their workload at the end of the semester:
 - “This was [the] most chilled of all projects [in the course].” (Fall 2020)
 - “Sprints [...] seemed to fly by, especially when the first few days is simply figuring out Trello/brushing up on what Scrum is.” (Fall 2020)

4. PITFALLS, LESSONS LEARNED, AND RECOMMENDATIONS

Developing and implementing this Scrum Boot Camp was an exciting learning experience for the instructors, as much as it was for students. Here are several take-aways and recommendations that may be useful to instructors trying to incorporate this type of project in their courses. The primary focus of this learning experience should be on the process (use

of Scrum artifacts, accountabilities, and events), rather than on the product (the scope and quality of actual solutions that students create). The pedagogical value is in helping students understand the importance of empiricism, flexibility, adaptability, constant communications, and transparency of the Scrum process. It provides the insight that a functioning product increment could be produced without a time- and resource-consuming multi-phase cycle, but it requires commitment, courage, focus, openness, and respect among the team members. Furthermore, the product does not need to be a fully completed solution by the end of Sprint 2; however, user stories in the Product Backlog and the Product Increments at the end of each Sprint should demonstrate the team's shared meaningful vision for the product evolution beyond Sprint 2.

Many students have difficulty thinking of their Product Backlog and Product Increments in terms of functional and nonfunctional requirements expressed as user stories (contained in Trello cards), and instead manage their product development in terms of specific tasks that must be performed. To help students address this common misconception, the instructor needs to monitor Product Backlogs, point out where a card is specified as a task and help students reframe the card as a user story (with appropriate tasks making up a checklist or a "definition of done"). It is very useful to constantly remind the POs that the user stories should be stated according to the template, such as "As a <user role>, I want to be able to <do what?> so that <why?>."

Several issues are related to students' misunderstanding of Scrum accountabilities. On many occasions, a student with an attractive product idea would claim the PO role but would soon realize that they want to develop the product and would be reluctant to delegate this to Developers while focusing on the Product Backlog. In other similar situations, such student would want to be a Developer, but for the user stories the team would look up to them, rather than to the PO. In one comical situation, a student asked whether she "could be all three" – the PO, a Developer, and the SM. While these are not uncommon occurrences, the instructor needs to make sure that students understand the trade-offs associated with taking on certain accountabilities and to steer students away from taking dual roles.

Another difficulty that students may have is understanding the SM as "a leader who serves" and the PO as a source of requirements. The presence of both accountabilities confuses students about "which of the two is the 'Project Manager' who tells us what to do." The instructor needs to explain to students that Scrum embraces servant leadership and collegial ownership rather than the administrative project management: every team member should be actively involved, take responsibility, and show initiative in generating, refining, and implementing ideas and overcoming any obstacles. Thus, the Scrum process values "individuals and interactions over processes and tools" (Dennis et al., 2010; *Manifesto for Agile Software Development*, 2001; Parris & Peachey, 2013; Sendjaya & Sarros, 2002).

Cultivating the SM accountability is a big challenge. Naturally, becoming a competent SM requires deep understanding of Scrum, training, and practice, all of which are premature to expect from students who are just being introduced to Scrum. Nevertheless, in the best-case scenario, a well-mixed and outgoing student, who, as a SM, can demonstrate a good sense of servant leadership, could be a great

example to other students and produce a strong pedagogical impact. In the worst-case scenario, a student would understand the SM role just as a responsibility to call and schedule the meetings.

Just like with any other student team projects, lack of engagement and "free-riding" do occasionally occur in Scrum Boot Camp, although, in our experience, more rarely than in other projects. On the one hand, this is not a very severe issue since teams are self-organized, and students who persistently demonstrated themselves as free riders in the previous projects find it difficult to join a team. Sometimes, several previous "free-riders" would form their own team in which there is no other way to succeed but to pull the weight, so there is a much higher chance they would get involved. On other occasions, a student with poor participation reputation may be welcomed into a team with a charismatic SM and dynamic members who would engage such a student. Computer-assisted peer evaluations are an efficient and reliable tool for the instructor to identify situations of students' adverse behaviors and intervene or adjust individual grades. Emphasizing and cultivating transparency in the teams through Daily Scrums and Sprint Retrospectives reduce adverse behaviors and engage students to help each other, although the lack of daily interactions and of full-time commitment to the team remain notable barriers.

Another frequent confusion in the student teams is "who manages the Trello Board?". Students are informed that the PO creates the Trello Board and curates the Product Backlog, yet any team member is welcome to take actions in the Board (with appropriate activity traces). It usually takes several repetitions to convey this idea, though. It is not unusual to hear this question from students even during Sprint 2, but commonly it is clear from Sprint Retrospectives that students realize the need for collegial ownership and transparent activity traces.

As was pointed out above, simulating Daily Scrums is one of the biggest challenges of running this project in a true Scrum manner. After some experimentation, we found the approach that worked for our course. Other instructors may have to discuss different options with students and experiment to see what works for them. It is paramount, however, to conduct "Daily Standups" to promote the culture of accountability and transparency within the Scrum process and not to dismiss them as unimportant "extras".

We found Trello to be a very suitable and convenient SaaS application for facilitating Scrum Boards. It is easy to set up and get started, and it provides full visibility of teams' activities to the instructor. Students are often astonished by the power of using a Kanban-style task and project management application and acknowledge that they wish they used it for other projects, too. Via "power-ups," Trello integrates well with other applications, such as email, calendar, cloud file storage, GitHub, and Slack. Trello's free plan is sufficient for the Scrum Boot Camp project; unfortunately, the number of free power-ups is limited, and students may not be able to experiment with all of them.

Finally, would the Scrum Boot Camp work with an online course delivery? The switch to online classes during the COVID-19 pandemic in 2020-2021 inadvertently provided an opportunity to experiment with virtual Scrum teams in a synchronous course format (with the same section size). To our surprise, not only were students able to successfully participate in this activity, but students' engagement in the events and the resulting Product Increments exceeded instructor expectations

than a typical face-to-face class (which, perhaps, warrants future investigation).

5. CONCLUSION

This teaching tip presents a hands-on active-learning experience that advances students' understanding of Scrum and may be applied within the constrained time frame of a conventional SAD or PM course. This learning experience is designed to facilitate a solid understanding of Scrum values, accountabilities, artifacts, and events; to expose students to the experience of using versatile SaaS tools for project management; to enable greater students' agency in self-managing teams and controlling their workload; and to stimulate interest in Scrum and encourage students to seek professional certification. The activity fits in the niche between a single-class activity and a semester project and is effective in both face-to-face and synchronous online courses. Our evidence suggests that this learning experience achieves its pedagogical objectives and results in a vastly positive student response.

6. ACKNOWLEDGMENTS

The author thanks the Editors and anonymous reviewers for their valuable comments and recommendations; JMU College of Business and colleagues at JMU X-Labs for supporting and encouraging this work; Paige Normand for inspiring the idea for the project; Dr. Thomas Dillon for coining the name "Scrum Boot Camp"; Dr. Amy Connolly, Dr. Jeffrey May, and Jim York, CST for valuable comments and suggestions for the paper; the Mobius SLIP and PeerLogic teams for their contributions; numerous students and graduates for their extensive feedback on how to improve this learning experience.

7. REFERENCES

Allan, G. (1999). Getting Students to Learn About Information Systems Project Management: An Experiment in Student-centred Learning. *Research in Post-Compulsory Education*, 4(1), 59–74. <https://doi.org/10.1080/13596749900200043>

Baars, W. (n.d.). The Dark Side of Agile. <https://www.projectmanagement-training.net/articles/the-dark-side-of-agile/>

Babik, D., Gehringer, E. F., Kidd, J., Pramudianto, F., & Tinapple, D. (2016). Probing the Landscape: Toward a Systematic Taxonomy of Online Peer Assessment Systems in Education. *CSPRED 2016: Workshop on Computer-Supported Peer Review in Education*. Educational Data Mining, Raleigh, NC.

Babik, D., Singh, R., Zhao, X., & Ford, E. (2017). What You Think and What I Think: Studying Intersubjectivity in Knowledge Artifacts Evaluation. *Information Systems Frontiers*, 19(1), 31-56. <https://doi.org/10.1007/s10796-015-9586-x>

Baham, C. (2019). Implementing Scrum Wholesale in the Classroom. *Journal of Information Systems Education*, 30(3), 141-159.

Bequette, B. (2002). *Process Control: Modeling, Design and Simulation* (1st edition). Pearson.

Brutus, S., & Donia, M. B. L. (2010). Improving the Effectiveness of Students in Groups with a Centralized Peer

Evaluation System. *The Academy of Management Learning and Education (AMLE)*, 9(4), 652-662.

Dennis, A., Wixom, B. H., & Tegarden, D. (2015). *Systems Analysis and Design: An Object-Oriented Approach with UML* (5th ed.). John Wiley & Sons.

Dennis, R., Kinzler-Norheim, L., & Bocarnea, M. (2010). Servant Leadership Theory. In D. van Dierendonck & K. Patterson (Eds.), *Servant Leadership: Developments in Theory and Research* (pp. 169–179). Palgrave Macmillan UK. https://doi.org/10.1057/9780230299184_14

Edison, S., Rehberg, B., Danoesastro, M., & Sherman. (2017, July 19). *Taking Agile Way Beyond Software*. BCG Global. <https://www.bcg.com/publications/2017/technology-digital-organization-taking-agile-way-beyond-software/>

Eriksson, U. (2016, April 6). 15 Awesome Free Resources to Learn About Agile & Scrum Methodology. *ReQtest*. <https://reqtest.com/agile-blog/15-awesome-free-resources-to-learn-more-about-agile-and-scrum-methodology/>

Finkel, S. (2016, April 15). *Scrum: Empirical Process Control vs. Defined Process Control*. Agile Aces. <https://www.agileaces.net/scrum-empirical-process-control-vs-defined-process-control/>

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.

González Sanchis, A. (2014, July 9). What SCRUM Is Not. *LinkedIn*. <https://www.linkedin.com/pulse/20140709212840-106642299-what-scrum-is-not/>

Gwosdz, M. M. (2020, June 29). Does Scrum Ruin Great Engineers or Are You Doing It Wrong? *Stack Overflow Blog*. <https://stackoverflow.blog/2020/06/29/does-scrum-ruin-great-engineers-or-are-you-doing-it-wrong/>

Jeffries, R. (2016, September 8). Dark Scrum [Blog]. *Ron Jeffries*. <https://ronjeffries.com/articles/016-09ff/defense/#fn:trans/>

Johnson, E. (2014, March 13). *Agile Is More than Just a Software Development Methodology*. <https://content.intland.com/blog/agile/agile-is-more-than-just-a-software-development-methodology/>

Kasperowski, R. (2014, September 3). Great Games for Scrum and Agile Learning. *Richard Kasperowski*. <https://kasperowski.com/great-games-for-scrum-and-agile-learning/>

Lending, D., Fulcher, K., Ezell, J. D., May, J. L., & Dillon, T. W. (2018). Example of a Program-Level Learning Improvement Report. *Research & Practice in Assessment*, 13, 34-50.

Mack, O., Khare, A., Krämer, A., & Burgartz, T. (Eds.). (2015). *Managing in a VUCA World* (1st ed. 2016). Springer.

Manifesto for Agile Software Development. (2001, February). Agile Manifesto. <https://agilemanifesto.org/>

Margetak, L. (2019, June 4). What Is Not SCRUM? *Medium*. <https://medium.com/swlh/what-is-not-scrum-b5c63670416a/>

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

- Name Line. (2019, August 9). *Ultimate Camp Resource*.
<https://www.ultimatecampresource.com/ice-breakers/name-games/name-line/>
- nTask. (2020, August 13). How to Successfully Defeat Dark Scrum in 2021? *NTask*.
<https://www.ntaskmanager.com/blog/dark-scrum/>
- Parris, D. L., & Peachey, J. W. (2013). A Systematic Literature Review of Servant Leadership Theory in Organizational Contexts. *Journal of Business Ethics*, 113(3), 377-393.
<https://doi.org/10.1007/s10551-012-1322-6>
- PaulS. (n.d.). Scrum. *TastyCupcakes.Org*.
<https://www.tastycupcakes.org/tag/scrum/>
- Petrova, S. (2019, January 18). Adopting Agile: The Latest Reports about the Popular Mindset. *Adeva*.
<https://adevait.com/blog/remote-work/adopting-agile-the-latest-reports-about-the-popular-mindset/>
- Rush, D., & Connolly, A. (2020). An Agile Framework for Teaching with Scrum in the IT Project Management Classroom. *Journal of Information Systems Education*, 31(3), 196-207.
- Saltz, J., & Heckman, R. (2020). Exploring Which Agile Principles Students Internalize When Using a Kanban Process Methodology. *Journal of Information Systems Education*, 31(1), 51-60.
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. ScrumGuides.org.
<https://www.scrum.org/resources/scrum-guide/>
- SCRUMstudy. (2017, September 20). *Daily Scrum: Why is it important?* <http://blog.scrumstudy.com/daily-scrum-why-is-it-important/>
- Sendjaya, S., & Sarros, J. C. (2002). Servant Leadership: Its Origin, Development, and Application in Organizations. *Journal of Leadership & Organizational Studies*, 9(2), 57-64. <https://doi.org/10.1177/107179190200900205>
- Sharp, J., Mitchell, A., & Lang, G. (2020). Agile Teaching and Learning in Information Systems Education: An Analysis and Categorization of Literature. *Journal of Information Systems Education*, 31(4), 269-281.
- Stuart, C. E. (2020). *Cynefin-Framework as a Guide to Agile Leadership: Which Project Management Method for Which Type of Project? - Waterfall, Scrum, Kanban?* BoD – Books on Demand.
- Sutherland, J., & Schwaber, K. (2007). Nut, Bolts, and Origins of an Agile Framework. *The Scrum Papers*, 224.
- Terry, J. (n.d.). *Basics and Benefits of Agile Method*. Planview.
<https://www.planview.com/resources/guide/agile-methodologies-a-beginners-guide/basics-benefits-agile-method/>
- Topi, H., & Spurrier, G. (2019). A Generalized, Enterprise-Level Systems Development Process Framework for Systems Analysis and Design Education. *Journal of Information Systems Education*, 30(4), 253-265.
- Vandersluis, C. (2014). Apply Agile Methodology to Non-software Enterprise Projects. *Project Management Institute*. PMI Global Congress, Phoenix, AZ, USA.
<https://www.pmi.org/learning/library/apply-agile-methodology-nonsoftware-enterprise-projects-9273/>
- Wischweh, J. (2019, February 27). The Agile Crisis – A Primer. *Medium*. <https://blog.usejournal.com/the-agile-crisis-2016-9fb1c2f52af5/>

AUTHOR BIOGRAPHY

Dmytro Babik received the Ph.D. degree in information systems from the University of North Carolina at Greensboro in 2015, the MBA degree from Tulane University, New Orleans, LA, in 2004, and MA degree in Economics from CERGE-EI, Charles University, Prague, Czechia, in 2002. Since 2015, he has been an Assistant Professor of Computer Information Systems at James Madison University, Harrisonburg, VA. His research interests include wicked problem solving in peer-review-based environments, social computing in open innovation, social learning analytics and data visualizations, and agile system development. Dr. Babik has published papers in *Information Systems Frontiers*, *IEEE Transactions on Learning Technologies*, and *Journal of Information Systems Education*, and presented at a number of conferences.



APPENDICES

Appendix A. Example of Pre-project Announcement Posted in Canvas

Heads up!

Scrum Boot Camp Project will formally start on Tuesday, _____, with an in-class activity.

Please carefully read project instructions. You must form your project team before Tuesday through Canvas. You also need to complete the following activity as you form your team. Do not worry – this activity will only involve thinking (to the best of your ability) and no deliverable.

Deep thinking about a problem is about 80% of successfully solving it. Therefore, taking this activity seriously will significantly help you succeed in the project.

Here it is: Before _____, I need you to do these three things:

1. Think about what you are really good at and like doing. What skills do you have that may be used to create something?
2. Think whether there is something that you have been really wanting to build but did not know how, did not have all the necessary skills, or did not find like-minded people who shared your interest in building it. Preferably, it has to be something of practical value to somebody and something that could not be created by one person in a few days or by several people in a day. Examples of such things include (but not limited to) a website, an app, a robot, an artwork, a book (or an e-book), origami, a science project, or a marketing campaign.
3. Discuss (1) and (2) with your to-be teammates and agree on what you all as a team would like to build as a product in this project. You may choose to build any product, tangible or digital, of any purpose or complexity. You will need to demonstrate requirements for this product and how you, as a team, implement these requirements in the product increments over two sprints. This means that your product would not have to be fully completed, but you will need to demonstrate two FUNCTIONING product increments.

Let me know (through the discussion board) if you have any questions.

Appendix B. Example of Project Instructions Posted in Canvas

Scrum Boot Camp Project Instructions

In this project, you will learn the essentials of the Scrum framework – accountabilities, artifacts, and events – by building a minimum viable product (MVP) in two one-week sprints. As soon as you have your project team put together, your team must complete the following steps:

ASAP:

- Form your team in Canvas under “Project Teams”;
- As you form your team, you should agree on what product you are going to build.

Before the first Sprint Planning:

It may look like a lot but should take only about 15-30 minutes to complete; Product Owners will be the busiest at this stage, but will enjoy the ride later :)

- Everyone in the team must refresh what Scrum is by reviewing the Projects Management and Scrum readings;
- Your team must agree who your team’s Product Owner will be; PO is responsible for gathering and formulating user stories (system requirements) and entering them in the Product Backlog, and overall, keeping the Trello Scrum Board in good order;
- Your team must agree who your team’s Scrum Master will be; SM is responsible for ensuring that the team collaborates towards successfully completing the project, facilitating Scrum events, ensuring that Scrum artifacts are in good order; communicating with the instructor;
- The rest of the team will be Developers responsible for building the product increments;
- Everyone in the team must sign up to trello.com (i.e., create their accounts); if you already have a Trello account, feel free to use it (I discourage you from creating another one just for this course);
- The Product Owner must create a new Team in Trello, name it T_YourTeamName, and invite all team members, as well as the instructor; the Product Owner should be the Administrator of the Trello Team, but should also make the Instructor the Administrator of the Team too (this is very important!);
- The Product Owner must create a new Board in Trello and name it B_YourTeamName;
- The Product Owner must add the Trello team to the Trello Board and notify the team members;
- The Product Owner must create the four following lists in your team’s Trello Board: Product Backlog, Sprint Backlog, In Progress, Done;
- The Product Owner must add to the Product Backlog several cards that describe features of your product; each card should have a short label indicating the feature; Description should contain a user story following this template “As a <USER ROLE>, I want to be able to <DO WHAT?> so that <WHY?>”; not all features need to be well-defined, but there should be a fairly sufficient number of cards that describe your product by Thursday.

On the day of the first Sprint Planning:

In class, your team will hold a Sprint Planning event (about 20 minutes) to:

- Discuss and mutually agree on the scope of work that is intended to be done during the Sprint;
- Select items (cards) from the Product Backlog that can be completed in one Sprint;
- Prepare a Sprint Backlog that includes the work needed to complete the selected Product Backlog items;
- Feel free to use labels, checklists, deadlines, assign team members, and other features of Trello.

On the day of the first Sprint Review and Retrospective, and the second Sprint Planning:

In class, your team will hold:

- a Sprint Review (5 minutes informal demo of your Product Increment to the class with Q&A);
- a Sprint Retrospective (10 minutes discussion of what went well and what needs to be improved in your team’s processes; team members only);
- Sprint Planning for the next sprint (about 20 minutes; team members only).

Before the second Sprint Review:

At home, enter your name in the team-member contribution evaluation assignment in Mobius SLIP.

On the day of the second Sprint Review and Retrospective:

In class, your team will hold:

- a Sprint Review (5-minute informal demo of your second Product Increment to the class with Q&A);
- a Sprint Retrospective (10-minute analysis of what went well and what needs to be improved in your team’s processes; team members only);
- team-member contribution evaluations for this project in Mobius SLIP.

Assessment:

This project is assessed based on evidence of Scrum accountabilities, events, and artifacts (including the Product Increment) (see the rubric), subject to individual adjustments at instructors' discretion.

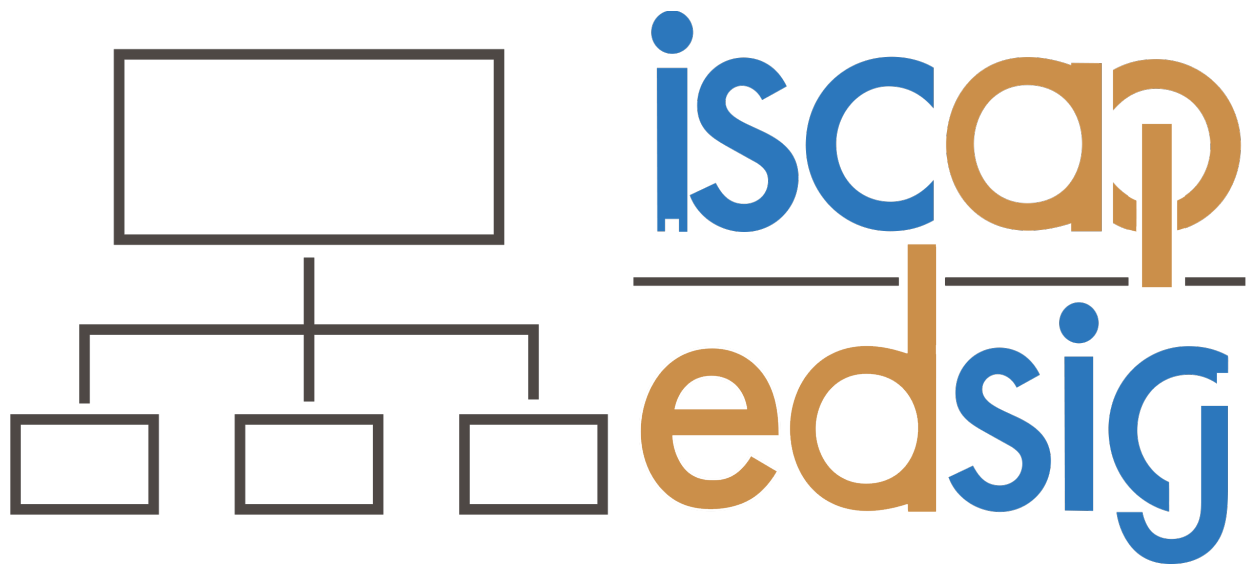
Missing team-member contribution evaluation will result in a 0 for the project.

If you receive an attainment index of 1 and certain keywords in reviews, you may receive 0 for the project.

Appendix C. Scrum Boot Camp Assessment Rubric

Criteria	Description	Points
Product Owner's activity	Clear evidence* of the Product Owner's activity and interactions with the Team	1
Scrum Master's activity	Clear evidence of the Scrum Master's activity and interactions with the Team	1
Meaningful Product Backlog	Product Backlog contains more user stories than would be implemented in two Sprints. User stories follow the specified format	1
Meaningful Sprint Backlogs	Sprint Backlogs for Sprints 1 and 2 are populated during the respective Sprint Planning events; adequate estimation (1 point per Sprint)	2
Sprint Review and Product Increment	Effective Sprint Review and meaningful Product Increment at the end of Sprints 1 and 2 (1 point per Sprint)	2
Sprint Planning	Clear evidence of effective Sprint Planning	1
Sprint Retrospective	Clear evidence of effective Sprint Retrospective	1
Daily Scrums	Evidence of effective Daily Scrums	1
Total Points		10

* Evidence for all criteria includes backlog items and activity records in the Trello Board, observed participation in team events (Planning, Review, and Retrospective sessions and standups), scores and comments in peer evaluations of team member contributions, and self-reflections.



**Information Systems & Computing Academic Professionals
Education Special Interest Group**

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.

Copyright ©2022 by the Information Systems & Computing Academic Professionals, Inc. (ISCAP). Permission to make digital or hard copies of all or part of this journal for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial use. All copies must bear this notice and full citation. Permission from the Editor is required to post to servers, redistribute to lists, or utilize in a for-profit or commercial use. Permission requests should be sent to the Editor-in-Chief, *Journal of Information Systems Education*, editor@jise.org.

ISSN 2574-3872