

Association for Information Systems

## AIS Electronic Library (AISeL)

---

International Research Workshop on IT Project  
Management 2019

International Research Workshop on IT Project  
Management (IRWITPM)

---

12-14-2019

# Challenges and Paradoxes of Teaching Project Management the Agile Way

Andreas Drechsler

Follow this and additional works at: <https://aisel.aisnet.org/irwitpm2019>

---

This material is brought to you by the International Research Workshop on IT Project Management (IRWITPM) at AIS Electronic Library (AISeL). It has been accepted for inclusion in International Research Workshop on IT Project Management 2019 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# Challenges and Paradoxes of Teaching Project Management the Agile Way

Andreas Drechsler  
Victoria University of Wellington  
andreas.drechsler.vuw.ac.nz

## ABSTRACT

This paper discusses challenges and paradoxes for teaching project management (PM) in an Agile way outside of a software development context. Based on a critical analysis of two PM course iterations in a professional masters program, the paper identifies several areas with tensions between established processes, norms, values, and expectations in higher education and the Agile PM course design. Ultimately, the paper finds that fulfilling the professional masters program's mission (to educate workforce-ready graduates for today's Agile / hybrid working environments) would require subverting numerous norms, values, and expectations on the course design, the students', and the lecturers' sides. Teachers and program directors in higher education can draw on this paper's findings to identify and manage pitfalls and paradoxes in their own PM course designs, in order to have them convey Agile PM's principles, values, and techniques effectively while retaining a positive student experience.

## Keywords Tables:

agile project management, agile education, project management education, agile paradoxes, teaching agile

## INTRODUCTION

The skill shortage in today's IT industry is not limited to software developers, but also includes roles without a coding background such as business analysts (Weston, 2019). Simultaneously, Agile<sup>1</sup> and hybrid project and organizational environments are becoming increasingly common in practice within and beyond the IT industry (Panditi, 2018; VersionOne, 2019). These hybrid environments are characterized by values, principles, methodologies, and techniques that fall somewhere between the extremes of a pure engineering-oriented approach (such as waterfall-oriented software development) and a pure Agile approach, and go by names such as Water-Scrum-Fall and others (Kropp et al., 2018; Kuhrmann et al., 2017; VersionOne, 2019; West, 2011).

To respond to these developments, a well-rounded project management (PM) education should therefore cover both traditional and Agile PM, as well as their respective advantages and disadvantages. Numerous papers report on conveying or embedding Agile in Higher Education (HE) courses (see "Teaching Agile in Higher Education" below) and related challenges and recommendations, but they rarely 'delve beyond the surface'. Moreover, most papers report on programming or software development courses. There, teaching Agile is comparably straightforward, since Agile stems from this area. However, PM courses are also part of HE programs without software development components. The challenge here is to find a suitable replacement domain for the application of Agile PM methodologies and techniques.

This paper contributes to filling these gaps by outlining a PM course design that aims to educate students on traditional as well as Agile PM, without drawing on any programming or software development. Based on the experiences in two iterations of the course with decidedly mixed results, this paper also problematizes (Alvesson and Sandberg, 2011) Agile teaching in HE by identifying tensions that arise from the Agile course design and teaching approach in a traditional HE context. These tensions may subsequently lead to challenges and paradoxes that students and lecturers encounter and have to navigate. This critical perspective on teaching Agile PM is informed by the author's experience and reflection and supporting quantitative and qualitative data that has been gathered throughout the two course iterations.

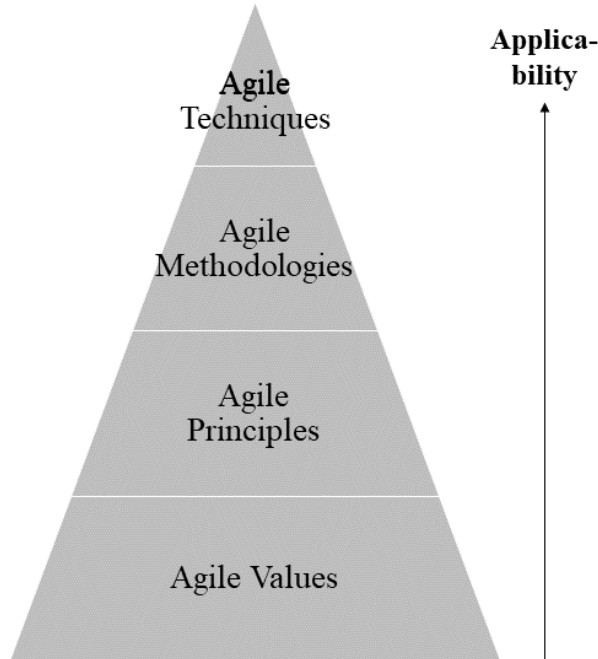
---

<sup>1</sup> For brevity, this paper uses the term Agile (with a capital A) as a term encompassing agile values, principles, methodologies, and techniques in general, without referring to specific ones (see also the second section).

**THEORETICAL AND CONCEPTUAL FOUNDATIONS**

**Traditional, Agile and Hybrid Project Environments**

Following Conboy (2009, p. 337), this paper defines Agility as the continuous readiness “to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment.” Beyond the definition, the Agile pyramid (Figure 1) allows distinguishing between Agile values, principles, methodologies, and techniques. The direct applicability increases towards the top of the pyramid, while the content of the layers below is retained or embodied. For instance, Agile methodologies offer a set of directly applicable techniques that are in-line with the Agile principles and values.



**Figure 1. The Agile pyramid (based on (AgileLion, 2019; Kropp and Meier, 2013) etc.)**

Table 1 lists the Agile values and principles based on the Agile Manifesto (Beck et al., 2001), which Tables 4 and 5 below will draw on.

<b>Agile Values</b>	<b>Agile Principles</b>
<p>“[W]e have come to value:</p> <ol style="list-style-type: none"> <li>1. Individuals and interactions over processes and tools</li> <li>2. Working software over comprehensive documentation</li> <li>3. Customer collaboration over contract negotiation</li> <li>4. Responding to change over following a plan</li> </ol> <p>That is, while there is value in the items on the right, we value the items on the left more.”</p>	<p>“We follow these principles:</p> <ol style="list-style-type: none"> <li>1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.</li> <li>2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.</li> <li>3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.</li> <li>4. Business people and developers must work together daily throughout the project.</li> <li>5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.</li> </ol>

	<ol style="list-style-type: none"> <li>6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.</li> <li>7. Working software is the primary measure of progress.</li> <li>8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.</li> <li>9. Continuous attention to technical excellence and good design enhances agility.</li> <li>10. Simplicity – the art of maximizing the amount of work not done – is essential.</li> <li>11. The best architectures, requirements, and designs emerge from self-organizing teams.</li> <li>12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.”</li> </ol>
--	--

**Table 1. Agile values and principles (Beck et al., 2001)**

Table 2 summarizes how Agile PM principles and values contrast with traditional PM. Note that Table 2 deliberately over-simplifies for the sake of illustration – any actual project setting in practice would probably fall somewhere between the two extremes.

	<b>Traditional PM</b>	<b>Agile PM</b>
<b>Project characteristics</b>	Well-understood, critical, stable scope	Exploratory / creative, non-critical, unclear / changing scope
<b>PM factors</b>	Time, Cost, Scope, Quality	Business value, the deliverable itself (time and cost are secondary)
<b>Role of the project manager</b>	Traditional (planning, directing, etc.)	No project manager
<b>Leadership style</b>	Command and control	Collaborative
<b>Decision-making style</b>	Directive / top-down	Cooperative
<b>Customer / user involvement</b>	Low	High
<b>Process emphasis</b>	High	Low
<b>Process changes</b>	Limited, exceptional	Common, encouraged
<b>Documentation</b>	High (explicit knowledge emphasized)	Low (tacit knowledge emphasized)
<b>Team member skills</b>	More specialized	More generalized / multidisciplinary
<b>Iterations</b>	Few, long duration	Many, short duration

**Table 2. Key characteristics of traditional, Agile, and hybrid PM environments (Fernandez and Fernandez, 2008; Špundak, 2014; Vinekar et al., 2006)**

### Teaching Agile in Higher Education

There are numerous papers that present specific course designs to teach Agile, often accompanied by experience reports, challenges, and recommendations (Alfonso and Botia, 2005; Anslow and Maurer, 2015; Babb et al., 2013; Budu, 2018; Burden and Steghöfer, 2019; Cubric, 2013; Kropp et al., 2014; Melnik and Maurer, 2003; Paasivaara et al., 2013, 2014; Schmitz, 2018; Schroeder et al., 2012; Steghöfer et al., 2016) – and this list is non-exhaustive by far. While an in-depth analysis of all these papers is not possible here due to space restrictions, a closer look reveals that almost all papers (with a few notable exceptions) cover the software development or programming domains, underlining the gap this paper intends to fill.

Notable papers that cover non-software domains are Cubric (2013), Pope-Ruark (2015), and Schmitz (2018). They use a wiki-based Agile Project Management encyclopedia, grant applications for non-profit organizations, and a PowerPoint presentation on a topic related to their major, respectively, as deliverables. The first two teaching approaches allow for a comprehensive and Agile student experience (from principles and values to selected

techniques), but at the expense of traditional PM content. Schmitz' (2018) approach is more intended as a primer to an in-depth coverage and experience of Agile, similar to other papers that report on using Agile simulation games (e.g. Scrum City Building with Lego bricks or cardboard) in software engineering courses (Kropp et al., 2014; Lynch et al., 2011; May et al., 2016; Paasivaara et al., 2014).

Moreover, while most papers discuss challenges and recommendations, these rarely go beyond 'single loop learning' (Argyris and Schön, 1995) regarding lessons learned for Agile PM course design. Moreover, all papers draw on reasonably effective and well-received courses. Course designs that were not effective (and thus would probably offer an increased potential for learning) are not reported on. Therefore, the current state of the literature indicates a need for deeper problematizations of teaching Agile, especially given the tensions one can infer from the differences between traditional and Agile PM environments as outlined in Table 2.

### A University Course and its Context – an Analysis Framework

Figure 2 shows the framework this paper will draw on for the critical analysis of an Agile PM course. The framework is based on the CIPOF (Context-Input-Process-Output-Feedback) framework for HE (Marshall, 2016, 2018) and expands and visualizes its elements.

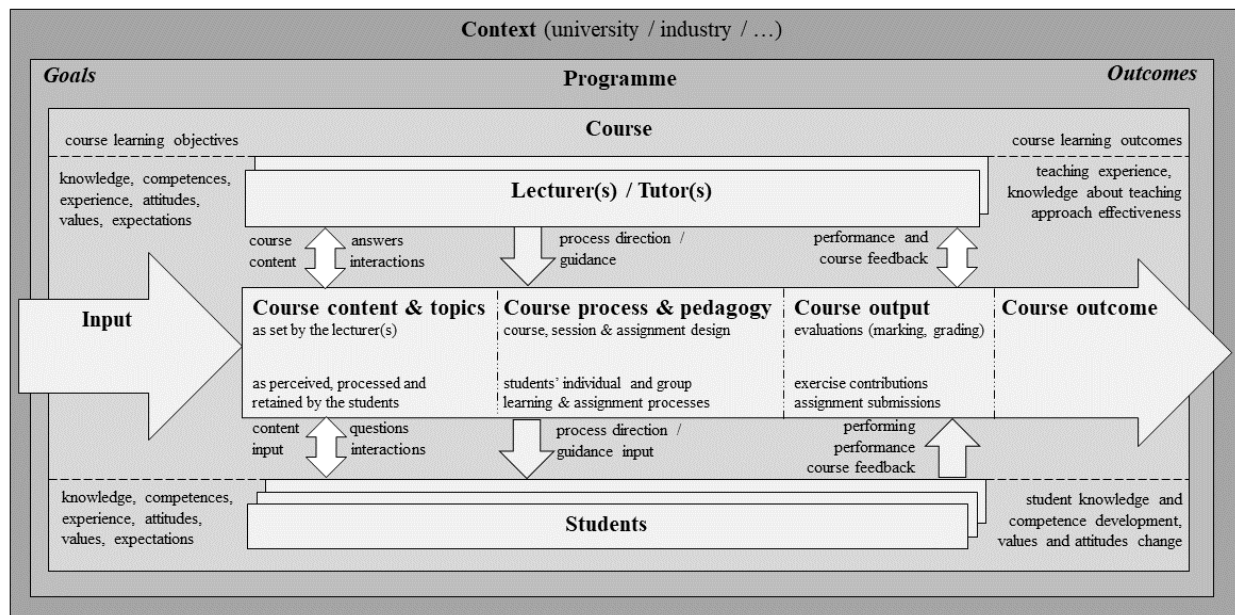


Figure 2. Analysis framework for the PM course

This framework embeds a course in its immediate context (a program of study) and wider contexts. These comprise at least the university – with aspects such as learning culture, core curriculum, or a graduate profile – as well as the industry where the program graduates would find employment. Each program has high-level goals it intends to reach (e.g., the graduate profile). The program-level outcomes (e.g., the graduates themselves) could be measured against these goals to assess a program's effectiveness. A course has corresponding course learning objectives which contribute to reaching the program's goals. A course consists of one or more lecturers (academic staff as well as (usually student) tutors) and the students enrolled in a course. They all bring their existing knowledge and competence to the course, but also certain attitudes, values, and expectations (e.g., regarding the teaching 'style' / learning approach, or the roles of students and teaching staff). The latter aspect is particularly relevant in this context since Agile teaching goes beyond content and technique learning, but also affects personal beliefs and values (Kropp et al., 2014). Relevant aspects of the learning process (depicted as the long white block arrow) are the 1) course content, 2) the corresponding pedagogical approach that determines how the content is delivered and learned (Sharp and Lang, 2018), and 3) the output (e.g., exercises, assignments), which ultimately leads to the 4) outcome (i.e., student development).

As not all pedagogical approaches and techniques are equally suited to teach all types of content (Mishra and Koehler, 2006), the course process and course content are interdependent to an extent. The suitability of content, process, and outputs in order to match the course goals and objectives and to achieve the intended outcomes is achieved through constructive alignment (Biggs, 1996) of these elements. Finally, constructivism (Hyslop-Margison and Strobel, 2007) highlights that each student perceives the offered content differently and constructs their own knowledge through their individual or group learning process based on their existing knowledge, competences, experiences, attitudes, values, and expectations.

## METHODOLOGY

The main means of developing the course design as well as the critical analysis presented in this paper were reflection-in-action and reflection-on-action (Schon, 1983). Since there were no easily adaptable course designs (due to software development activities most often being very much ‘ingrained’ in them, see above), the author (who designed, coordinated and taught in the course) based the course design and its evolutions on his own interpretation and application of Agile values and principles to teaching Agile PM in an Agile way. These considerations included how they could help achieve specific PM-related learning objectives along the levels of Bloom’s taxonomy (Krathwohl, 2002). The author also reflected on possible or actual implications before, during and after the courses were offered (hence reflection both in and on-action). The reflections to develop the subsequent critical analysis took place in two modes: D-reflection (deconstruction, defence, declaiming, destabilizing, danger-warning) and R-reflection (reconstruction, reframing, reclaiming, re-presentation) (Alvesson et al., 2008). In a nutshell, D-reflection aims to uncover hidden assumptions and perspectives and challenge them – possibly fundamentally. In contrast, R-reflection seeks constructive engagement through alternative perspectives, in order to develop new theoretical, empirical, political, or ethical contributions. Used dialectically, D-reflection can provide a necessary foundation for subsequent R-reflection. This paper emphasizes D-reflection over R-reflection to problematize Agile teaching.

To gain a comprehensive complementary picture from the student perspective, the author’s reflection activities were supported by quantitative and qualitative data that was gathered during and after the two course offerings. A number of weeks after the final grades were awarded for each course iteration, the entire student cohort was invited to take part in a short survey with the intention of measuring the students’ perception of the viability of the Agile teaching approach, their self-assessed familiarity with the Agile principles and values, and their self-assessed prowess in using Agile methodologies and techniques – all through a 5-point Likert scale. The quantitative part of the survey was supported by open-ended course feedback questions. In addition, the students’ learning blogs (see next section) were analyzed regarding notable insights related to the Agile teaching approach (with a single exception where a student asked that their blog entries would not be used for this purpose).

## A COURSE DESIGN FOR TEACHING AGILE PM THE AGILE WAY

Figure 3 gives a detailed depiction of the course design for the BUAN567 (course code is anonymized) project management course.

The course context is a 1-year professional conversion masters program with the intent to produce workforce-ready junior business analysts (BA). Students do not require prior work experience to enter the program but need to have a non-IT undergraduate degree. BUAN567 takes place in the second trimester, and the objective is to acquaint the students with traditional as well as Agile PM so that they can work as BAs in traditional, Agile, or hybrid project environments. The course runs over 12 weeks with a single session per week. The first week is devoted to an introduction into PM and team-building exercises. Weeks 11 and 12 cover topics beyond PM such as organizational change management, portfolio management as well as a course recap. Each session runs from 9:30am to 2:30pm with a lunch break. The whole course time can be freely allocated to lectures, exercises, assignment work, etc. The course has been co-taught by the author and one industry professional, each one being responsible for about half the sessions.

The dilemma on how to balance traditional and Agile PM content, and what type of a meaningful deliverable to choose was resolved as follows: The two major *assignment deliverables* are a traditional business case and project plan, with several sessions (held by the industry professional) covering the corresponding content. To deliver these assignments, the students are placed in teams and are asked to follow an *Agile process*. Week 2 acquaints them with the Scrum methodology via a ball game simulation (May et al., 2016) conducted by two experienced Agile facilitators from the industry, and gets them started on the assignment work by guiding them through a first short Scrum sprint and some essential Scrum techniques. Other Agile principles, techniques, and tools are covered in subsequent weeks by the

academic lecturer in one of his three roles. The other two roles are 1) role-playing a client from a fictional company the students are already acquainted with from previous courses – a coffee shop chain – and 2) to serve as process guide (‘Scrum Master Master’) for the Agile team processes upon request.

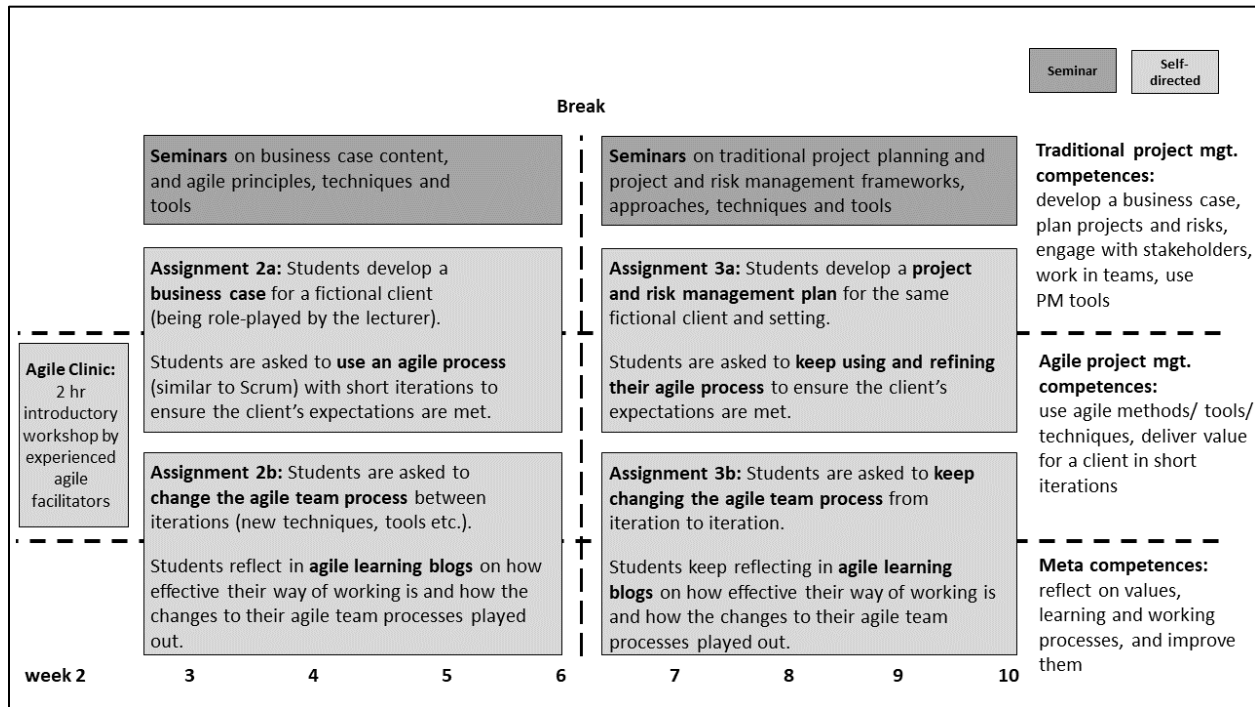


Figure 3. PM course design overview (Drechsler, 2018)

For their assignments, the students are asked to develop a business case (up to week 6) and a subsequent project plan (up to week 11) for the role-played client to improve a selected issue in the client’s fictional company – such as increasing the coffee roasting process efficiency or the communication effectiveness between cafes and the central kitchen. The students are free to arrange meetings or to conduct sprint reviews at any time with the fictional client or send e-mails / Slack messages to gather more information. To distinguish his three roles, the lecturer wears one of three baseball caps during the interactions with the student teams outside the regular classroom sessions: one with the university’s logo for the traditional lecturer role, one with a logo close to the fictional company for the role-played client, and one with a rugby team logo for the ‘Scrum Master Master’ role (since Scrum is originally a rugby play).

To achieve a sustained balance between content (business case and project plan) and the Agile process, to ensure regular changes to each team’s Agile process, and to foster student reflection on Agile values and principles, the students are also asked to write private journals (‘Agile blogs’, with four entries per trimester half) on their experiences and the changes to the Agile team process. This course design therefore addresses three levels of competence development: traditional PM, Agile PM, and meta-competences (Bogo et al., 2013) such as reflection and continuous change and improvement. Tables 3 and 4 show in detail how this course design implements the Agile principles and values (cf. Table 1).

At the time of writing, the course has been offered two times. Most assignment deliverables were assessed to be good to excellent in both iterations, with the second iteration producing a considerably higher customer satisfaction (with exceptions) which is reflected in a higher overall grade distribution. However, both the course evaluation and the additional quantitative post-course surveys show decidedly mixed results in terms of the students’ reception and their perception of their own self-efficacy in applying Agile. Here, the second iteration resulted in considerably worse scores than the first (Table 5).

<b>Agile values</b>	<b>in the course design</b>
1. Individuals and interactions over processes and tools	While the students are exposed to traditional and Agile PM methodologies, techniques, and tools, the emphasis lies on the interactions within the student teams and with the role-played customer to produce the two main deliverables. The student teams decide for themselves how they want to support their interactions with suitable techniques and tools. Moreover, each student gets their own individual voice through their Agile learning blogs.
2. Working software over comprehensive documentation	Since the two deliverables (business case and project plan instead of software) are developed for the fictional customer (and the marking rubric reflects this), the emphasis in the student assignment work therefore lies on producing content that is valuable for a customer.
3. Customer collaboration over contract negotiation	The close collaboration with the fictional customer is intended to be one of the main cornerstones of the students' assignment work. The assignment briefs have just a supporting function.
4. Responding to change over following a plan	Since the customer may not appreciate every part of the deliverable drafts he is reviewing, the assignment work especially towards the end becomes responding to customer feedback instead of merely filling out a business case or project plan template. Moreover, the teams are very much encouraged to reflect on and change their way of working to improve their team processes (otherwise there would not be much to blog about).

Table 3. Agile values in the course design

<b>Agile principles</b>	<b>in the course design</b>
1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	The importance of frequent reviews of draft versions of the deliverables with the fictional customer is emphasized from week 2 onwards, and the marking rubrics highlight the importance of satisfying the customer expectations, and not to 'surprise' him.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	The lecturer role-playing the fictional customer can make sure that there are always changing requirements, and can simultaneously tailor the extent of the changes to team performance and the time remaining.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	Since the time to deliver each deliverable is roughly four weeks, the need for short iterations is implied through the need for frequent reviews (principle 1) and to write four Agile learning blog entries over the course of these four weeks.
4. Business people and developers must work together daily throughout the project.	The lecturer role-playing the fictional customer aims to respond to student messages (e-mails and Slack) as quickly as possible, and to be available for face-to-face meetings and reviews as necessary.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	The course design assumes that students are motivated to achieve a good grade, and the teams are given high autonomy how they want to work on the deliverables.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	This principle is emphasized early in the course, and the fictional customer will actively suggest face-to-face meetings if e-mails do not seem to be effective.



7. Working software is the primary measure of progress.	Instead of working software, showing draft versions of the deliverables to the fictional customer are the main means to measure progress during the assignment work and to receive feedback for the next iteration.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	The most challenging principle to implement, since each student in the program takes the same two other courses at the same time, and some may have other responsibilities on top of that as well. To allow each team the highest possible freedom how they organize themselves (including the sprint length) there are no set milestones etc. for reviews of intermediate versions of the deliverables.
9. Continuous attention to technical excellence and good design enhances agility.	Continuous attention to technical excellence is achieved through regular customer feedback since the fictional customer also gives feedback that a lecturer would give (just in a different guise). For the second course iteration, the industry professional was also available to give feedback to project plan drafts concerning the drafts' 'technical excellence', and essentially acted as a Senior PM / PMO representative to support the teams' efforts to develop an excellent project plan.
10. Simplicity – the art of maximizing the amount of work not done – is essential.	One idea of frequent customer interactions is for the student teams to understand the customer needs early on and to tailor the business case and project plan outlines and document depth accordingly, instead of merely filling out templates and ending up with content the customer may be less interested in. The fictional company is small to medium sized and therefore both deliverables can be quite 'barebones' and still satisfy the customer expectations.
11. The best architectures, requirements, and designs emerge from self-organizing teams	The student teams have high autonomy for how they want to organize themselves.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly."	The importance of sprint retrospectives is highlighted in week 2, and the requirements for the Agile learning blog content emphasize the necessity to write about one's reflection of the team process and the changes the team agreed to make between iterations.

**Table 4. Agile principles in the course design**

While there were a number of seemingly minor changes between the first and the second iteration (a larger course with more teams, a new lecturer from industry, more exposure to specific Agile techniques, a new Kanban simulation, some timetabling reshuffling due to lecturer availability, ...), the overall course design largely stayed the same. The author's interpretation of the substantial differences in reception is that the second course iteration brought a number of underlying tensions between the Agile teaching approach and existing norms, values, and expectations to the fore that may have been already 'bubbling under the surface' the first time round. The subsequent section will therefore explore the identified tensions and subsequent paradoxes of Agile teaching in the HE context in greater depth.

Question	I found it helpful to experience the agile principles and values through working on course assignments in an agile fashion	I found it helpful to reflect on my agile experiences in the agile blogs	I was able to influence how my team organized its agile way of working	I found it helpful to be able to influence how my team organized its agile way of working	I feel now well-acquainted with agile principles and values	I feel now well-acquainted with agile techniques and the Scrum method	I acquired meaningful agile-related skills for my future as a BA	I feel confident in applying an agile approach in the future	I found it sensible to run a part of the course assignments in an agile manner	
Avg	2018	1.45	2.55	2.09	2.18	1.55	1.73	1.82	2.45	1.82
	2019	3.38	3.62	3.31	3.08	2.54	2.92	3.23	3.15	3.46
	Diff	1.93	1.07	1.22	0.9	0.99	1.2	1.41	0.7	1.64
Std. Dev	2018	0.5	1.16	0.79	0.83	0.66	0.62	0.72	0.89	0.72
	2019	1.15	1.27	1.2	1.27	1.01	1.14	1.19	1.03	1.28
	Diff	0.65	0.12	0.41	0.44	0.35	0.52	0.47	0.14	0.56

**Table 5. Results of the quantitative post-course surveys**  
(n<sub>2018</sub>=11 of 22 students; n<sub>2019</sub>=13 of 32 students; scale 1-5, lower is better)

## TENSIONS AND PARADOXES WHEN TEACHING PM THE AGILE WAY

This section contrasts traditional teaching approaches through the Agile lens, and then discusses tensions and resulting paradoxes of the Agile teaching approach following the main building blocks of a course (cf. Figure 2): the lecturer, the course design, and the students. Due to space restrictions, the goal for this section is to provide an identification and a first discussion of the tensions and paradoxes. Following Putnam et al. (2016, p. 72), a paradox is understood as “[c]ontradictions that persist over time, impose and reflect back on each other, and develop into seemingly irrational or absurd situations because their continuity creates situations in which options appear mutually exclusive, making choices among them difficult.”

### Contrasting Traditional Teaching Approaches with the BUAN567 approach

Table 6 contrasts traditional teaching approaches with the specific Agile teaching approach employed for BUAN567. Note that Table 6, like Table 1, over-simplifies the traditional approach and treats it as one extreme point of a continuum for illustrational purposes. Many actual teaching settings will deviate from what Table 6 depicts. Table 6 covers a course’s main building blocks to achieve the course outputs (and subsequent outcomes): the lecturer roles, the two course design aspects of content and process, and the student roles (cf. Figure 2).

In sum, Table 6 shows that there is a fundamental shift taking place for all course building blocks when moving from a traditional teaching approach to the Agile teaching approach for the BUAN567 course. While Tables 3, 4, and 6 underline the high alignment of the BUAN567 course design to the Agile spirit, the supporting data shows that this approach was less than well-received especially in its second iteration (Table 5). The following subsections explore challenges and paradoxes that are introduced by these fundamental shifts in all course building blocks in the wider context of the values and expectations of a course in a HE context.

### Tensions and Paradoxes on the Lecturers’ Side

Shifting the lecturer roles to de-emphasize traditional teaching in favor of the combination of multiple roles as outlined in Table 6 created tensions since the new set of roles defies existing role expectations from the students, resulting in comments along the line of “*You didn’t teach us!*” (paraphrased from the formal course evaluation). While the traditional lecturer role of preparing ‘bite-sized’ ready-to-consume lecture content was indeed and intentionally largely unfulfilled, the learning process shifted to the self-directed assignment work in close collaboration with the customer (see also next section). As mentioned before, the outcomes were good to excellent in both iterations, as especially the higher overall grades in the second iteration show. In other words, the role shift led to the paradox that intended learning outcomes on all three levels (traditional PM, Agile PM, and meta-competences) were demonstrably reached, while there were impressions among the students that learning (in the traditional sense) had not taken place.

	<b>Traditional teaching approaches</b>	<b>BUAN567 Agile teaching approach</b>
<b>Lecturer roles</b>	<ul style="list-style-type: none"> <li>• ‘Sage on the stage’</li> <li>• ‘Showrunner’ (course coordinator)</li> </ul>	<ul style="list-style-type: none"> <li>• ‘Guide on the side’ <ul style="list-style-type: none"> <li>○ Facilitate the seminars, give and point towards relevant PM content</li> <li>○ PM coach (‘Senior BA / PM’)</li> <li>○ Process coach (‘Scrum Master Master’)</li> </ul> </li> <li>• Fictional client (role-played)</li> <li>• ‘Showrunner’ (course coordinator)</li> </ul>
<b>Course content</b>	<ul style="list-style-type: none"> <li>• Lecture content is prepared and offered in-line with the course learning objectives</li> <li>• Lecture content clearly defines the scope of a course</li> <li>• Lecture content is presented sequentially (‘waterfall-ish’)</li> </ul>	<ul style="list-style-type: none"> <li>• Course content is presented and offered to enable the students to <ul style="list-style-type: none"> <li>○ deliver a business case and a project plan to fulfil the fictional customer’s expectations</li> <li>○ inform and foster change in their way of working on the deliverables</li> </ul> </li> </ul>
<b>Course process</b>	<ul style="list-style-type: none"> <li>• Assignments follow lecture content (again, ‘waterfall-ish’) and may even explicitly refer to specific content</li> <li>• Assignment briefs outline the goals and the process on how to work on the assignment deliverables</li> <li>• Assignment deliverables are marked based on rubrics related to lecture content and learning objectives</li> </ul>	<ul style="list-style-type: none"> <li>• Course and assignment scope are in part negotiated during the process of engaging with the fictional client</li> <li>• The process of producing the assignment deliverables is equally important as the course content</li> <li>• Assignment deliverables are marked by rubrics related to customer satisfaction</li> </ul>
<b>Students</b>	<ul style="list-style-type: none"> <li>• <i>consume</i> tailor-made lecture content</li> <li>• <i>produce</i> related assignment work with a stable scope</li> <li>• <i>receive</i> grades based on clear rubrics</li> <li>• are <i>masters</i> of their own work process</li> <li>➔ Knowledge and competence development are at the center of a course</li> </ul>	<ul style="list-style-type: none"> <li>• are <i>presented and offered</i> industry-standard materials as inputs to consider for their assignment work</li> <li>• <i>produce</i> client-oriented work with a flexible scope</li> <li>• <i>receive</i> grades based on client satisfaction...</li> <li>• ... and how they <i>manage and change</i> their own work processes and attitudes</li> <li>➔ Incorporates changes to values and belief systems</li> <li>➔ Knowledge and competence development happen as part of the assignment work</li> </ul>

**Table 6. Contrasting key characteristics between traditional and Agile teaching approaches**

A second paradox regarding the lecturer roles is that the sole unchanged ‘showrunner’ / course coordinator role stayed the same in the new format, simply due to this being a formal requirement from the surrounding university context. This role (and also the role of a marker giving a numerical score for each assignment submissions, to an extent) served as a regular reminder (or throwback) for everyone to the traditional style of how a course is run, every time an organizational matter had to be taken care of or a numerical mark had to be given for an assignment, punctuating the Agile experience in a sense. Even the role name ‘lecturer’ (as it is officially called in the author’s university) contrasts with the Agile teaching approach (cf. Table 6) and thus continuously reinforces this tension.

### **Tensions and Paradoxes regarding the Course Design (Content and Process)**

A second source for tension is the course design itself, where the process of working on the assignment deliverables ultimately drives the learning process. The effects of the shift as shown in Table 6 becomes best apparent when analyzing the course design through three of the four elements of the iron quadrangle: time, scope, and quality (cost is excluded due to irrelevance). Quality is the major criterion driving the student marks for their assignment work.

Traditionally, an assignment's scope is lecturer-driven and stable throughout the course, while the student is the 'master' of controlling how much time they want to invest for assignment work, and when they want to do it. Students are also in control over the level of quality they aim for, and striving for the best quality is ultimately down to the student skill in 'guessing' the lecturer's expectations conveyed in class and the assignment briefs (akin to traditional requirements engineering for IT projects having to 'guess' what the customers actually want).

In the BUAN567 course design, both scope and time now essentially become client-driven, however. The actual scope of an assignment appears somewhat opaque at first for the students since they first encounter only industry-standard templates or examples for business cases and project plans. They would need one or more sessions with the fictional client before being able to adjust and reduce the scope according to the client expectations. The necessary time needed to fulfill the client expectations is now variable, since time now essentially depends on how well the initial drafts of each deliverable are received by the client, and how quickly and how well the client feedback is incorporated into subsequent drafts. Since the deliverables are produced in a group effort, the need to coordinate the group work exacerbates the tensions generated by scope and time now being opaque factors outside the teams' control. Along these lines, quality now essentially becomes a function of the number of iterations with the client until all feedback is incorporated and no more room for improvement is left.

To exaggerate, the resulting marks do not anymore reward an individual *being good at guessing what the lecturer wants* but now reward the team *willing to invest the most time, focus, and effort for iterations and being able to cope best with the additional uncertainty*. Coupled with every team member essentially managing their own portfolio of projects (assignments) across several courses, this paradoxical effect is neither aligned with the course learning objectives nor with the reality of the program. As an example for how the data analysis and reflection processes took place, Table 7 summarizes them for this particular paradox.

Issue	Client-driven assignment process / time as a variable
<b>Student perception (blog entry)</b>	<i>"We didn't actually have any face to face meetings or communications in this sprint, it was all completed over email or group chat forums. It's not ideal doing things this way but this was largely a product of the time pressure we found ourselves under – each of us had our own outside demands on our time so we had to find a way to communicate and complete the work remotely. I think it did probably have a slightly negative impact on the overall quality of our work; we didn't get that valuable time to chat issues through and get group consensus [...]."</i>
<b>Student perception (survey)</b>	Especially in the second course iteration, the students report that they had only limited influence on their agile way of working, and they did not find it very sensible to run course assignments in an agile manner (Table 5). In addition, in the formal course evaluation for the second course iteration, more than two thirds of the respondents indicate that the course workload was too much or far too much.
<b>Author / lecturer perception</b>	Once a student team embraces the agile values and principles around frequent client interaction, client satisfaction and responding to change, and thus sends a number of drafts of the deliverables for feedback, the team's effort to achieve actual customer satisfaction and responding to the requested changes may well be considerable, even if the lecturer in the role of the fictional client chooses to give feedback concentrating on only the most essential issues.
<b>D-reflection</b>	Designing the assignment process in a client-centric way essentially takes away control over the assignment process from the student and transfers it to the role-played client. This shift subverts existing norms, values and expectations in HE and also the practicalities of being part of a program with a number of concurrent courses and assignments.
<b>Paradox</b>	Assigning marks based on customer satisfaction in a client-driven Agile assignment process ultimately rewards the team willing to invest the most time, focus, and effort for iterations and being able to cope best with the additional uncertainty due to the loss of control.

**Table 7. From data and perception to reflection and paradox identification**

Moreover, the HE context with fixed course outlines and prescriptions does not allow a truly Agile reaction in terms of making sweeping changes to assignments halfway through a course once such issues become apparent. Larger-

scale changes to a course are only possible between offerings (akin to project changes between portfolio cycles, for instance). In a sense, the Agile course design is still stuck in a non-Agile ‘corset’ of the surrounding HE environment.

### **Tensions and Paradoxes on the Students’ Side**

A third source of tension lies on the student side. The previous subsections discussed several ways how student expectations, values, and attitudes towards a HE course are subverted by the chosen Agile teaching approach. While it was mentioned before that a change in values and attitudes is part of teaching Agile, it was also mentioned that not all subversions are in line with the overall intentions behind the course design or practical in the reality of the actual course offering.

Moreover, a strong customer-orientation is at the core of Agile. Viewed through an Agile lens, in the university context the students themselves are the *customers* – they are paying to acquire knowledge and skills so that they become employable as BAs, and they are also paying for a good experience during the program. In the course context, however, the students are simultaneously the *products* (a product with agency, so to speak), as developing and changing their knowledge, competence, and values is the whole point of the course. Together with the far-reaching subversions of expectations, values, and attitudes, this duality leads to the paradox that the expectations, values, and attitudes of students in the customer role clash with the experiences of the students in the product role when the course design subverts the former set of expectations, values, and attitudes.

### **Overarching Paradoxes and Challenges**

Taken together, the paradoxes discussed in the previous three subsections culminate in two more overarching paradoxes: While the Agile course design subverted values and expectations for how a university course ‘works’ on several fronts, it was simultaneously being very much in-line with 1) Agile values and principles (cf. Tables 3 and 4) and 2) the program’s overall mission (to educate graduates ready for Agile / hybrid working environments). These two paradoxes mirror to an extent the challenges of Agile project teams in traditional or hybrid organizational environments.

The overarching challenge when teaching Agile the Agile way is now to cope with the identified tensions and paradoxes. While developing practical solutions for doing so lies outside this paper’s scope, one approach could be to take into account the course building blocks outlined in Figure 2 and Table 6, and decide as part of the overall course design where to stay within traditional boundaries, and where to ‘dial up’ the extent of being Agile. Making the tensions, paradoxes, and subversions of role expectations and values explicit to the students could also contribute to a more successful and well-received course design. Such an approach could be further supported by placing a stronger emphasis on those Agile techniques that can identify, address, and mitigate these paradoxes and their consequences.

### **DISCUSSION, CONCLUSION, OUTLOOK**

This paper contributes a course design for a PM course that covers traditional as well as Agile PM and does not depend on software development skills for the Agile part. The course design goes beyond ‘doing Agile’ by not only incorporating Agile methodologies and techniques but also trying to convey the spirit of ‘being Agile’ through a more holistic Agile learning process embodying Agile principles and values. Against the backdrop of the mixed course evaluation results, the paper further contributes a problematization of this additional step towards ‘being Agile’ in teaching by identifying several tensions and paradoxes that the particular chosen way of teaching Agile incurs in a traditional HE context. These tensions and paradoxes cut across all course building blocks of the lecturers, the students, and the course design itself. These tensions illustrate that a true shift towards ‘being Agile’ in HE teaching is fraught with the overarching challenge of addressing these latent paradoxes before they become salient. Moreover, as long as there is a non-Agile context surrounding a course, any course trying to follow a ‘being Agile’ approach will always be confined to being effectively a hybrid model, based on the restrictions, requirements, and expectations imposed by its context.

Reflecting further on these findings, many course designs in the literature that teach Agile methodologies and techniques within otherwise quite traditional settings may well have intentionally or unintentionally sidestepped many of the tensions and paradoxes discussed above. In the author’s interpretation, the tensions and paradoxes only came to the fore in the BUAN567 course design due to the design decision to go the crucial step further in trying to truly ‘be Agile’, instead of merely ‘doing (some) Agile’ in an otherwise traditional HE course setting. Some of the

situational factors that changed between the first and second iteration of the course may have been contributing factors to having the tensions and paradoxes negatively affect the course process and the student perception.

HE lecturers and program directors wanting to truly ‘walk the talk’ when teaching Agile can draw on the tensions and paradoxes discussed in this paper to design their course around them and, in other words, remain on the ‘doing Agile’ side of things in teaching to emphasize effectiveness and manageability of a course. They can also choose to deliberately steer their course design towards selected tensions and paradoxes, and tackle them front and center by introducing suitable Agile techniques, and couple this with an explicit intended learning outcome of how to navigate Agile paradoxes in hybrid environments (which is another worthwhile skill for graduates to have).

When considering these implications, however, one has to take this paper’s limitations into account. First, the paper covers only one particular Agile PM course design of many possible, and also has a limited foundation of only two course offerings. Second, there is a strong subjective component of reflection and interpretation by the author regarding the tensions and paradoxes, albeit supported with quantitative and qualitative data. Space restrictions for this paper meant that there was also little room to engage further with the available data. Lastly, the paper emphasizes the identification and initial discussion of the tensions and paradoxes over a more in-depth treatment and theorization of each tension and paradox, or a discussion of ideas how to cope with them in future course designs.

Further research can therefore further investigate, discuss and theorize the identified tensions and paradoxes. A second obvious avenue for further research is to adapt the course design in response so that it still delivers a truly Agile course while demonstrably coping with the identified tensions and paradoxes, leading to an overall more well-received course and student experience, and while still providing a balanced education in traditional and Agile PM.

## REFERENCES

- AgileLion (2019) Agile Pyramid (<http://agilelion.com/tags/agile-pyramid>, accessed August 15, 2019).
- Alfonso, M. I., and Botia, A. (2005) An Iterative and Agile Process Model for Teaching Software Engineering, in *18th Conference on Software Engineering Education Training (CSEET'05)*, 9–16 (<https://doi.org/10.1109/CSEET.2005.5>).
- Alvesson, M., Hardy, C., and Harley, B. (2008) Reflecting on Reflexivity: Reflexive Textual Practices in Organization and Management Theory, *Journal of Management Studies*, 45, 3, 480–501.
- Alvesson, M., and Sandberg, J. (2011) Generating Research Questions Through Problematization, *Academy of Management Review*, 36, 2, 247–271 (<https://doi.org/10.5465/amr.2009.0188>).
- Anslow, C., and Maurer, F. (2015) An Experience Report at Teaching a Group Based Agile Software Development Project Course, in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education, SIGCSE '15*, New York, NY, USA: ACM, 500–505 (<https://doi.org/10.1145/2676723.2677284>).
- Argyris, C., and Schön, D. A. (1995) *Organizational Learning II: Theory, Method, and Practice*, Reading, MA: Addison Wesley.
- Babb, J. S., Hoda, R., and Nørbjerg, J. (2013) Barriers to Learning in Agile Software Development Projects, in *International Conference on Agile Software Development*, Springer, 1–15.
- 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., and Thomas, D. (2001) Manifesto for Agile Software Development (<http://www.agilemanifesto.org>, accessed October 3, 2012).
- Biggs, J. (1996) Enhancing Teaching through Constructive Alignment, *Higher Education*, 32, 3, 347–364.
- Bogo, M., Katz, E., Regehr, C., Logie, C., Mylopoulos, M., and Tufford, L. (2013) Toward Understanding Meta-Competence: An Analysis of Students’ Reflection on Their Simulated Interviews, *Social Work Education*, 32, 2, 259–273 (<https://doi.org/10.1080/02615479.2012.738662>).
- Budu, J. (2018) Applying Agile Principles in Teaching Undergraduate Information Technology Project Management, *International Journal of Information and Communication Technology Education*, 14, 3, 29–40 (<https://doi.org/10.4018/IJICTE.2018070103>).
- Burden, H., and Steghöfer, J.-P. (2019) Teaching and Fostering Reflection in Software Engineering Project Courses, in *Agile and Lean Concepts for Teaching and Learning: Bringing Methodologies from Industry to the Classroom*, D. Parsons and K. MacCallum (eds.), Singapore: Springer Singapore, 231–262 ([https://doi.org/10.1007/978-981-13-2751-3\\_12](https://doi.org/10.1007/978-981-13-2751-3_12)).
- Conboy, K. (2009) Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development, *Information Systems Research*, 20, 3, 329–354.

- Cubric, M. (2013) An Agile Method for Teaching Agile in Business Schools, *The International Journal of Management Education*, 11, 3, 119–131 (<https://doi.org/10.1016/j.ijme.2013.10.001>).
- Drechsler, A. (2018) *Teaching Agile in Business Analyst Education: Course Design and First Experiences*, presented at the 9th annual conference of Computing and Information Technology Research and Education New Zealand (CITRENZ), Wellington, NZ.
- Fernandez, D. J., and Fernandez, J. D. (2008) Agile Project Management—Agilism versus Traditional Approaches, *Journal of Computer Information Systems*, 49, 2, 10–17.
- Hyslop-Margison, E. J., and Strobel, J. (2007) Constructivism and Education: Misunderstandings and Pedagogical Implications, *The Teacher Educator*, 43, 1, 72–86.
- Krathwohl, D. R. (2002) A Revision of Bloom’s Taxonomy: An Overview, *Theory into Practice*, 41, 4, 212–218.
- Kropp, M., and Meier, A. (2013) Teaching Agile Software Development at University Level: Values, Management, and Craftsmanship, in *2013 26th International Conference on Software Engineering Education and Training (CSEE&T)*, 179–188 (<https://doi.org/10.1109/CSEET.2013.6595249>).
- Kropp, M., Meier, A., Anslow, C., and Biddle, R. (2018) Satisfaction, Practices, and Influences in Agile Software Development, in *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*, ACM, 112–121.
- Kropp, M., Meier, A., Mateescu, M., and Zahn, C. (2014) Teaching and Learning Agile Collaboration, in *2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE&T)*, 139–148 (<https://doi.org/10.1109/CSEET.2014.6816791>).
- Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., Trektre, K., McCaffery, F., Linssen, O., and Hanser, E. (2017) Hybrid Software and System Development in Practice: Waterfall, Scrum, and Beyond, in *Proceedings of the 2017 International Conference on Software and System Process*, ACM, 30–39.
- Lynch, T. D., Herold, M., Bolinger, J., Deshpande, S., Bihari, T., Ramanathan, J., and Ramnath, R. (2011) An Agile Boot Camp: Using a LEGO®-Based Active Game to Ground Agile Development Principles, in *2011 Frontiers in Education Conference (FIE)*, F1H-1-F1H-6 (<https://doi.org/10.1109/FIE.2011.6142849>).
- Marshall, S. (2016) Quality as Sense-Making, *Quality in Higher Education*, 22, 3, 213–227 (<https://doi.org/10.1080/13538322.2016.1263924>).
- Marshall, S. J. (2018) Quality as Sense-Making, in *Shaping the University of the Future: Using Technology to Catalyze Change in University Learning and Teaching*, S. J. Marshall (ed.), Singapore: Springer Singapore, 315–347 ([https://doi.org/10.1007/978-981-10-7620-6\\_15](https://doi.org/10.1007/978-981-10-7620-6_15)).
- May, J., York, J., and Lending, D. (2016) Play Ball: Bringing Scrum into the Classroom, *Journal of Information Systems Education*, 27, 2, 87–92.
- Melnik, G., and Maurer, F. (2003) Introducing Agile Methods in Learning Environments: Lessons Learned, in *Extreme Programming and Agile Methods - XP/Agile Universe 2003*, Lecture Notes in Computer Science, F. Maurer and D. Wells (eds.), Springer Berlin Heidelberg, 172–184.
- Mishra, P., and Koehler, M. J. (2006) Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge, *Teachers College Record*, 108, 6, 1017–1054.
- Paasivaara, M., Heikkilä, V., Lassenius, C., and Toivola, T. (2014) Teaching Students Scrum Using LEGO Blocks, in *Companion Proceedings of the 36th International Conference on Software Engineering*, ACM, 382–391.
- Paasivaara, M., Lassenius, C., Damian, D., Rätty, P., and Schröter, A. (2013) Teaching Students Global Software Engineering Skills Using Distributed Scrum, in *2013 35th International Conference on Software Engineering (ICSE)*, 1128–1137 (<https://doi.org/10.1109/ICSE.2013.6606664>).
- Panditi, S. (2018) Survey Data Shows That Many Companies Are Still Not Truly Agile, *Harvard Business Review* (<https://hbr.org/sponsored/2018/03/survey-data-shows-that-many-companies-are-still-not-truly-agile>).
- Pope-Ruark, R. (2015) Introducing Agile Project Management Strategies in Technical and Professional Communication Courses, *Journal of Business and Technical Communication*, 29, 1, 112–133 (<https://doi.org/10.1177/1050651914548456>).
- Putnam, L. L., Fairhurst, G. T., and Banghart, S. (2016) Contradictions, Dialectics, and Paradoxes in Organizations: A Constitutive Approach, *The Academy of Management Annals*, 10, 1, 65–171 (<https://doi.org/10.1080/19416520.2016.1162421>).
- Schmitz, K. (2018) A Three Cohort Study of Role-Play Instruction for Agile Project Management, *Journal of Information Systems Education*, 29, 2, 93–104.
- Schon, D. A. (1983) *The Reflective Practitioner: How Professionals Think in Action*, New York: Basic Books.
- Schroeder, A., Klarl, A., Mayer, P., and Kroiß, C. (2012) Teaching Agile Software Development through Lab Courses, in *Proceedings of the 2012 IEEE Global Engineering Education Conference (EDUCON)*, 1–10 (<https://doi.org/10.1109/EDUCON.2012.6201194>).

- Sharp, J. H., and Lang, G. (2018) Agile in Teaching and Learning: Conceptual Framework and Research Agenda, *Journal of Information Systems Education*, 29, 2, 45–52.
- Špundak, M. (2014) Mixed Agile/Traditional Project Management Methodology–Reality or Illusion?, *Procedia-Social and Behavioral Sciences*, 119, 939–948.
- Steghöfer, J.-P., Knauss, E., Alégroth, E., Hammouda, I., Burden, H. akan, and Ericsson, M. (2016) Teaching Agile: Addressing the Conflict Between Project Delivery and Application of Agile Methods, in *Proceedings of the 38th International Conference on Software Engineering Companion*, ICSE '16, New York, NY, USA: ACM, 303–312 (<https://doi.org/10.1145/2889160.2889181>).
- VersionOne (2019) 13th Annual State Of Agile Report (<https://explore.versionone.com/state-of-agile/13th-annual-state-of-agile-report>, accessed August 12, 2019).
- Vinekar, V., Slinkman, C. W., and Nerur, S. (2006) Can Agile and Traditional Systems Development Approaches Coexist? An Ambidextrous View, *Information Systems Management*, 23, 3, 31–42.
- West, D. (2011) *Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today* (<http://varunm.com/wp-content/uploads/2015/04/water-scrum-fall.pdf>).
- Weston, M. (2019) Top, in-Demand Roles for 2019 Revealed, *The HR DIRECTOR*, (<https://www.thehrdirector.com/business-news/employment/top-demand-roles-2019-revealed909/>, accessed August 30, 2019).