Using Agile Games to Invigorate Agile and Lean Software Development Learning in Classrooms

Rashina Hoda

SEPTA Research, Department of Electrical and Computer Engineering
The University of Auckland, New Zealand

Abstract A wide variety of professional certifications, trainings and dedicated academic courses are attempting to meet the ever-growing demand for software professionals competent in the knowledge and use of agile and lean software methods and practices. Agile games, embodying experiential learning, are popular in industrial contexts and are increasingly being trialed in academic settings as a feasible alternative or a complement to traditional instructional learning approaches. Most games reported, however, focus exclusively on the Scrum method and practices. This study reports on the use of four agile games for learning fundamental agile and lean concepts such as iterative and incremental delivery, collaborative estimation, pair programming, and work-in-progress limits. Based on classroom observations and survey-based quantitative and qualitative data, we found that: agile games are a useful supplement for effective learning, can easily invigorate learner engagement and promote team-building. Effective facilitation and debriefing sessions are imperative to the success of agile games in classrooms. Educators can easily introduce agile games by selecting from a variety of accessible online resources based on their ability to deliver desired learning outcomes and graduate attributes to invigorate learning about agile and lean software development.

Keywords Agile software development, games, engagement, learning, teaching, classroom

1 INTRODUCTION

Agile and lean software development has witnessed widespread adoption in the software industry (Deemer et al., 2010) as well as in academia over the past two decades (Scott et al., 2016). Rapid response to frequent changes in requirements, fast delivery times, close customer collaboration, empowered self-organizing teams, and adaptive management are some of the distinguishing features that have accelerated the adoption and practice of agile methods (Schwaber & Beedle, 2002; Beck, 1999; Hoda et al., 2013, Augustine, 2005). The industry's demands for software professionals trained and competent in agile and lean methods and practices is met by efforts from several professional trainers and universities offering dedicated courses on these topics (Kruchten, 2011; Wangehiem et al.,

2013). The use of capstone projects to introduce agile methods has been a popular approach in such agile courses (Mahnic, 2012; Lu & DeClue, 2011; Schroeder et al., 2012; Scott et al., 2016). The theoretical concepts underlying agile and lean software development, however, still need to be imparted regardless of whether the course includes a practical component. This is typically achieved through traditional classroom instructional approaches such as lectures (Wangeheim et al., 2013, Mahnic, 2012). Making the information-laden component of the course engaging and effective is nontrivial.

Games – embodying experiential learning – are viewed as a feasible alternative or a complement to traditional instructional learning approaches (Percival et al., 1993; Wangeheim & Shull, 2009). Games are seen to offer a unique opportunity to leverage high user-engagement and interactivity potential combined with learning objectives (Wouters et al., 2009; Sylvester et al., 2014); and provide motivation for learning (Malone, 1981) as players often display deep concentration and imbibe concepts through gameplay (Shneiderman, 2004).

Games for learning Agile software development concepts, termed in this chapter as 'Agile games', are widely prevalent in industrial training contexts and many games are easily accessible online (e.g. Heintz, 2016; Cohn, 2016; Kerievsky, 2016; Kniberg, 2016). The efficacy and ability of games to support agile learning is increasingly being trialed in academic contexts (Fernandes & Sousa, 2010; Wangeheim et al., 2013; Paasivaara et al., 2014). Games such as PlayScrum (Fernandes & Sousa, 2010), Scrumia (Wangeheim et al., 2013) and those using Lego®-bricks (Lynch et al., 2011; Paasivaara et al., 2014), designed to simulate the Scrum lifecycle, have reported evidence of successful use in academic settings. Most of these games, however, have focused on one particular agile method, Scrum, simulating the entire Scrum development cycle or a particular Scrum practice. Other concepts and practices of agile software development such as XP's popular pair programming have received relatively little attention. Similarly, Lean and Kanban concepts have remained relatively untouched in reported academic contexts.

This study involved the use four games – *Paper Planes* (Heintz, 2016), *Planning Poker* (Cohn, 2016), *Pair Draw* (Kerievsky, 2016), and the *Name Game* (Kniberg, 2016) – in a graduate-level, dedicated agile and lean software development course at the University of Auckland. These games primarily focus on concepts such as: agile software development's *iterative and incremental delivery* model; Scrum's *collective team estimation* practice; XP's *pair programming* practice; and Kanban's *work-in-progress limit* concept respectively. They were selected from a plethora of available online Agile games based on their ability to map to the desired learning outcomes and graduate attributes of the course, as described later.

Based on classroom observations and qualitative and quantitative data collected through a specially designed survey, our main findings are that: the four Agile games effectively supplemented learning of fundamental agile and lean concepts; the games strongly invigorated classroom engagement; and promoted team building. Some of the lessons learned from the experience include: effective facilitation was found to be valuable to the efficient use of games in the classroom; and that a dedicated debriefing session was imperative for each game to help

establish connections between the game experience and the underlying theoretical concepts and instigate discussions and insights.

2 BACKGROUND AND RELATED WORKS

2.1 Agile and Lean Software Development

Agile software development (Highsmith & Fowler, 2001) is an umbrella term that captures a set of values and principles that guide the modern, light-weight software development processes such as Scrum (Schwaber & Beedle, 2002) and eXtreme Programming (Beck, 1999). Among its core values are: people and interactions; customer collaboration; working software as a measure of progress; and embracing change. Emerging in the late 1990s, agile methods have rapidly become the default software development method of choice in the global software industry (Deemer, Benefield, Larman, & Vodde, 2010).

Along similar lines and originating from the Toyota Product System (Taiichi, 1988; Womack et al., 1990), is the concept of Lean software development. Manifested in concrete methods such as Kanban (Anderson, 2010), Lean propounds the values of eliminating waste, optimizing the whole, building in quality, constant learning, fast delivery, engaging everyone, and continuous improvement (Poppendieck, 2003). Lean values are adopted independently or often alongside agile methods to finetune the software development process.

There are several fundamental concepts related to agile and lean methods available in accessible formats (Deemer et al., 2010.) The four key concepts that are most relevant to this research study and were used in the course are described below.

- **Iterative and incremental delivery:** is one of the defining features of agile software development where iterative refers to the short two-to-four-week work-cycles within which a team develops a prioritized sub-set of product features; and incremental refers to the continuous integration of these features to form a complete product over the course of the project. Such an approach is in stark contrast to the traditional sequential development and delivery models which were marked by specific stages of design, development, testing, and deployment (Royce, 1970). In agile methods such as Scrum a selected set of requirements move through all the above stages and are delivery-ready by the end of every iteration (Schwaber & Beedle, 2002). The concept of such fixed time-boxed iterations, however, is not a part of lean methods such as Kanban which focus more on continuous workflow (Wang et al., 2012.)
- **Effort estimation:** is the practice of estimating the effort involved in producing a customer requirement expressed in the form of a user-story and often representing a particular product feature; including design, development, and

testing. A unique aspect of effort estimation in an agile or lean context is that it not only involves project managers or technical leads but rather the whole team including developers and testers (Highsmith & Fowler, 2001). Since the user stories are typically self-assigned later in the iteration, estimation of these stories in the early planning stages needs to involve the whole team so that everyone can discuss and understand the requirements and can provide their input into the estimation based on experience (Hoda et al., 2013.)

- **Pair programming:** is one of the most well-known XP practices which involves two developers working together on a single computer to develop software features together, where one writes the code, referred to as the driver, and the other is meant to provide guidance and help, referred to as the navigator (Beck, 1999). The partners swap roles frequently. Pair programming continues to be an important and popular agile practice (Williams et al., 2010).
- **Work-in-Progress (WIP) limit:** is the concept of placing constraints on the number of work items being executed at any given time with the aim to reduce multi-tasking and improve productivity. The WIP limit ensures that a pull system is followed and nothing is built before it is needed (Wang et al., 2012). Kanban's focus on limiting WIP differentiates it from most agile methods.

Following the rapid growth and popularity of agile and lean methods worldwide, many professional training courses, workshops, and certificates are available (e.g. Scrum Master certification by the Scrum Alliance, Agile Certified Practitioner by the Project Management Institute). Several multi-national companies are dedicated to producing tools and offering services to support agile and lean software teams (e.g. VersionOne, Rally Software, ThoughtWorks).

Similar trends have been witnessed in the education domain where courses dedicated to agile and lean software development are offered by many computer science and software engineering departments in universities around the world e.g. University of Oxford, Carnegie Mellon, John Hopkins, Alto University, UNICEN University. While it is obvious that knowledge and experience in agile and lean methods is beneficial to the job prospects of current and future graduates, the approaches to teaching and learning of these methods varies. Using capstone projects to allow students to practice and experience agile methods has been widely reported (Mahnic, 2012; Lu and DeClue, 2011; Schroeder et al., 2012; Scott et al., 2016.) While our course SOFTENG761 at the University of Auckland includes a quasi-real-world project experience component as described later, where the delivery of the fundamental agile and lean concepts prior to commencing the project in an engaging and effective manner was a challenge.

2.2 Learning Agile and Lean through Games

Games have been described as goal-directed, competitive activities with agreed rules (Lindley, 2004; Wouters et al., 2009). Games whose main purpose is other

than entertainment such as for learning and instruction are referred to as serious or educational games (Wouters et al., 2009.) Educational games often involve simulations and role-playing which are seen to embody experiential learning (Kolb, 1984).

There is a growing interest in the use of games for learning as they offer a unique opportunity to leverage high user engagement and interactivity potential combined with learning objectives (Wouters et al., 2009; Sylvester et al., 2014). Furthermore, games are seen to provide motivation for learning (Malone, 1981) as players often display deep concentration and imbibe concepts through gameplay (Shneiderman, 2004).

Since agile and lean methods focus on people and interactions, collaborative games harness as well as promote these skills. Furthermore, agile methods propound rapid response and games provide hands-on opportunities to experience and promptly react to situations and contexts. It is therefore no surprise that games are popular means of teaching and learning agile concepts in professional training contexts and are being increasingly trialed in academic settings.

2.2.1 Industrial Offerings

In industrial training contexts, simulations, role-play, and games-based learning have been actively used (e.g. certification courses offered by the Scrum Alliance). A plethora of resources are freely available online to support game-driven approaches for use by industrial professionals and academics alike. The games relevant to our study include: Paper Planes (Heintz, 2016), Planning Poker (Cohn, 2016), Pair Draw (Kerievsky, 2016), and the Name Game (Kniberg, 2016). Complete descriptions and instructions for each of these games are available on their respective websites. Brief descriptions of the purpose, execution, and expected outcomes of the four games are presented below to aid the understanding of the research context and results to follow.

- ***Paper Planes:*** is a collaborative simulation game used to introduce agile software development. It primarily focuses on the iterative and incremental delivery model and provides the opportunity to experience teamwork, customer collaboration and reflective practice. It involves multiple teams working on creating paper planes within time-boxed intervals or iterations. The facilitator explains the rules of the game and keeps time. They also act as the customer during the simulation, providing requirements and conducting acceptance tests of delivered product, for example all planes must have blunt heads for safety and those not complying are rejected. The simulation typically includes three iterations and lasts around two hours. Each iteration is divided into a 'plan-do-check' cycle where the plan stage involves the team planning their iteration and lasts one minute, the do stage involves actual production of the paper planes and lasts two minutes, and the check stage involves reflection as a team and lasts one minute. Teams are typically able to understand the importance of planning and

reflection as their productivity expectations change from being over-inflated to realistic over the course of the simulation. Different versions of this popular game are available online along with details of materials required, play instructions, and game resources (Heintz, 2016.)

- ***Planning Poker:*** is a collaborative gamified estimation technique used for effort estimation in Scrum (Cohn, 2016). Instead of estimating real user stories, in a simulated, training context, the learner teams are asked to estimate the sizes of various items, such as animals on a scale of one to ten, where one is the smallest and ten represents the largest. A photo of an animal is presented and individuals are asked to think of a size rating in their mind without sharing or discussing with others. On the count of three, all learners are asked to show their rating by raising equivalent number of fingers (or cards printed with numbers). The facilitator then goes up to each team and singles out the individuals with the maximum difference in estimations, e.g. one and four, for the same animal and asks them to discuss their rationale. The idea is to allow team members to hear and understand other's perspectives and potentially refine their estimation based on the new information. The discussions continue until the gap is closed or lessened. This is repeated for other animals. As new information becomes available to teams, they may decide to recalibrate their previous estimations. The simulation enables individuals to appreciate multiple perspectives gained during collaborative estimation and learn to think and estimate as a team as opposed to individuals. The game usually takes around an hour including time for debriefing. Various ratings scales include sequential numbers, the Fibonacci series, or T-shirt sizes (extra-small, small, medium, large, extra-large) and others.
- ***Pair Draw:*** (Kerievsky, 2016) is game which simulates XP's pair programming practice. First, the learners are asked to individual draw a face on a sheet of paper. The drawings can be as simple or as detailed as they prefer. Then the facilitator asks the learners to form pairs (having an even number of learners helps). The exercise is repeated but this time, the pairs need to coordinate with each other to draw a single face collaboratively. Individuals within a pair are asked to use different coloured pens so it is easy to tell which bits were drawn by whom. Pairs can share their experiences and drawings with the wider group. The simulation is followed by a debrief session where the facilitator highlights the similarities between the simulation and the actual pair programming practice. Other questions are also posed to the group for discussion, such as what was the difference in drawing alone and drawing as a pair? What were the advantages and disadvantages of each? These allow the learners to form some expectations of what actual pair programming may entail.
- ***The Name Game:*** (Kniberg, 2016) is a simulation game that helps illustrate how multitasking can reduce efficiency and productivity. It is used to enforce the concept of applying work-in-progress (WIP) limits in Kanban. It involves team members playing the role of customers providing their names while one of the team members plays the developer who writes the names down. Two iterations of the game are played. In the first iteration, each customer provides a single letter of their name at a time to the developer who writes it down on the

customer's piece of paper and then attends to the next customer. This continues until all customers are served (i.e. their complete names are written.) This scenario represents multitasking as the developer simultaneously works on all customer requests. The second iteration involves the same task but the approach is different. Here the developer attends to one customer at a time, writing their full name down before moving on to the next customer. In other words, a WIP limit of one customer request at a time is applied. It is usually noticed that the second scenario (with WIP limit in place) yields better productivity and generally better experiences for the developer and customers.

2.2.2 Academic Adoption

In the academic context, the use of games and simulations for imparting software engineering concepts has been reported (Drappa and Jochen, 2000; Baker, 2005; Carrington et al. 2005.) The use of games for learning agile and lean concepts has also been trialed in various contexts. Fernandes & Sousa (2010), Lynch et al. (2011), Wangeheim et al. (2013), and Paasivaara et al. (2014) form a growing body of evidence in support of using games for teaching and learning agile concepts.

PlayScrum – a card game devised for use in the university context (Fernandes & Sousa, 2010). The focus of the game was to provide players with the experience of the Scrum method and the role of the Scrum Master. Feedback from thirteen Masters-level students who trialed the game found it be to visual, simple, useful, and enjoyable. The authors noted that the game was best used to complement the teaching of the theoretical concepts in class.

Wangeheim et al. (2013) presented and used Scrumia – a physical paper-based game, as a low-budget approach to complement more the traditional classroom instruction of Scrum. It involved simulating a scrum development cycle in teams of six people. The study provided evidence that the game provided a positive effect on learning Scrum. Like Fernandes & Sousa (2010), Wangeheim et al. (2013) also propound that such games are effective in complementing theoretical lectures.

The use of Lego®-based games has also been explored in academic settings (Lynch et al., 2011; Paasivaara et al., 2014.) Lego® bricks were used to build physical products in an agile boot camp setting involving learning across three iterations (Lynch et al. 2011.) Students showed a preference for the agile boot camp over lecture-based teaching. Paasivaara (2014) used a Lego®-based simulation game in a Masters level course at Alto University. Surveys of learner experiences showed high levels of satisfaction. While Lynch et al. (2011) found no difference in the recall of concepts introduced through games and those in lectures, Paasivaara et al. (2014) reported gains in learning and insight as a result of gameplay.

As explained above, numerous low-budget, paper-based games originating from industry are easily accessible online while some have originated from the academic domain (Fernandes & Sousa, 2010; Wangehiem et al., 2013). Most of these games have focused on one particular agile method – Scrum – and aimed to simulate the entire Scrum development cycle or a particular Scrum practice. Other concepts and

practices of agile software development such as XP's popular pair programming have received relatively little attention. Similarly, lean and Kanban concepts have remained relatively untouched in academic contexts.

Our research study reports on the use of four Agile games – Paper Planes, Planning Poker, Pair Draw, and the Name Game – for learning agile software development's iterative and incremental delivery, Scrum's collective team estimations, XP's pair programming and Kanban's WIP limits concept respectively, in an academic context and provides evidence to support the games-driven approach for agile and lean learning.

3 RESEARCH CONTEXT AND DESIGN

3.1 The Course Context

SOFTENG761 – *Agile and Lean Software Development* is course offered to final year bachelor of engineering (BE) and masters of engineering studies (ME studies) learners specializing in software engineering in the Electrical and Computer Engineering department at the University of Auckland. Like most courses at the university, this is a 12-week course with a two-week break typically separating two parts of six weeks each. The ratio of BE final year and the ME studies learners tends to vary between three to two (3:2) and equal (1:1). The course was first designed and launched by the author who has continued run the course in the capacity of the course director and lecturer. It has since witnessed consistent interest and growth, reflecting the growth of the software engineering program. Learners enrolling into the course are expected to possess advanced programming skills in popular languages such as Java, and have prior academic or industrial experience of team-based software projects. Their advanced skills and experience is expected to help them select from a variety of projects to implement, requiring a multitude of latest technologies and tools.

The main purpose of the course is to teach: the fundamental concepts of agile and lean software development such as iterative and incremental software development, self-organizing teamwork, customer collaboration, and project management; and the core practices from the Scrum, XP, and Kanban methods such as release and iteration planning, pair programming, and team estimations. The teaching approach for SOFTENG761 is based on a three-tier learning approach comprising of three progressive parts focusing on:

- ***Part I - Theory:*** provides learners with the fundamental knowledge about agile and lean software development over the first three weeks, typically imparted in the form of direct lecture-based instruction using PowerPoint slides,

supplemented by videos and classroom discussions. Theory is tested at the end of week three in a test worth 25%.

- ***Part II - Practice:*** provides learners with the opportunity of hands-on practice in agile and lean software development. Learners are required to self-form teams and select from the project options offered by local software companies. Students typically form teams of seven to eight students each. The projects cover a six to seven week period following weekly iterations and hands-on practice of several Scrum, XP, and Kanban practices. This makes up 50% of the assessment. Further details of the projects and the industry collaboration aspects of this course are beyond the scope of this chapter.
- ***Part III - Research:*** requires learners to search and review current research literature on various agile and lean-related topics with a focus on critically analysing these concepts in light of literature and their own project experiences and presenting them as an individual research essay worth 25%.

3.2 The Challenge

Delivering fundamental knowledge on any subject requires the definition and description of several new terminologies, concepts, and ideas. In the earliest offering of the course, this was achieved through the instructor talking through PowerPoint slides to a classroom full of learners with occasional questions and answers. In a bid to make things more interactive, learners were presented with scenarios from software development contexts and encouraged to discuss and debate possible outcomes relevant to agile practices. The main challenges associated with this approach were: level of learner engagement remained average as only some learners participated in the question-answer sessions or discussions around contexts while a majority of the class remained in passive recipient mode. As such it was unclear to what extent the class was engaged and whether the learners effectively comprehended the concepts being covered.

3.3 Introducing Games

Games were introduced into the course to improve the level of engagement and learning in the theory part of the course. A set of four different Agile game (games for learning Agile software development), were identified and added to the instructional materials and design. These were selected based on their ability to align to the course's desired learning outcomes and set of graduate attributes listed by the Washington Accord (Accord, 2013), an international agreement between bodies responsible for the accreditation of tertiary level engineering qualifications. Employing a set of graduate attributes defined by the Accord, such as *WA09 individual and team work*, provides a valuable framework for achieving

constructive alignment with learning outcomes during design of engineering courses. A mapping between the games, learning outcomes, and graduate attributes is presented in Table 1.

Table 1. Mapping of Collaborative Games deployed, underlying Agile and Lean Concepts, Course Learning Outcomes, and Graduate Attributes defined by the Washington Accord (Accord, 2013).

Agile Games	**Underlying Agile and Lean Concepts**	**Learning Outcomes**	**Most Relevant Graduate Attributes***
Paper Planes	Iterative and incremental software delivery	LO 1. Plan for project release and iterative delivery. LO2. Develop, test, and deliver software in an iterative and incremental fashion.	WA09, WA10, WK04, WK06, WA11
Planning Poker	Collective team estimations	LO3. Estimate user stories.	WA09, WA10, WK04, WK05, WK06
Pair Draw	Pair programming	LO4. Work effectively in a self-organizing team environment.	WA09, WA10, WK04, WK06
Name Game	Work-in-Progress (WIP) Limit	LO5: Design strategies to overcome common challenges of Agile and Lean practices.	WA09, WA10, WK04, WK06, WP06

* WA09: Individual and Team Work; WA10: Communication; WK04: Specialist Knowledge; WK05: Engineering Design; WK06: Engineering Practice; WA11: Project Management and Finance; WP06: Conflicting Stakeholder Requirements.

As the games were played prior to the projects commencing, learners from the same project teams were asked to play together. All games were played as project teams, except for Pair Draw where learners were asked to form pairs within teams. Brief descriptions of these games have been provided earlier in section 2.3 and complete details are available on their respective websites as referenced.

The games were played in classroom sessions following a brief introduction to the relevant concepts via PowerPoint slides and in-class discussions. Two-hour sessions were found to be necessary and sufficient for including some instructional material and at least one game. However, the Paper Planes game took up the entire two hours as it involved several steps and iterations. The games were facilitated by the lecturer/author with the help of a teaching assistant in charge of time keeping. The lecturer/author is an agile researcher and a certified Scrum Master with wide experience of teaching agile and lean concepts and practices. The lecturer moved around the entire classroom, encouraging learners through questions about their approach and progress. All learners were seen to actively engage in the games.

Further details of classroom experiences are discussed and shown later in the Findings section.

3.4 Data Collection and Analysis

Human ethics approval was gained prior to collecting data from the learners. The game sessions were observed by the lecturer and the teaching assistant. Notes were taken to record these observations. Quantitative and qualitative data was collected from learners about their experience of playing the games in the course through a specially designed games survey. The games survey was created and hosted on Google Forms and was conducted soon after the last game was played in class and the theory part of the course was over, in week four. The form was setup to require user authentication using the university credentials to ensure only learners from the course participated, however the details of the participants were not stored to allow for anonymous and more candid responses. The form was further set to accept only one response per person. A link to the survey was emailed out to the class and participation was completely voluntary. Twenty-three of the total 51 learners (45%) responded to the survey.

The games survey included a set of questions about games for learning in general and about the experience of playing specific games in class (see full list in Appendix.) The questions were of three types: multiple choice questions (MCQs), ratings using Likert scale (1-5) type questions, and open-ended questions, for example:

- **[MCQ]** The number of games used in SOFTENG761 was: Too few/Just right /Too many
- **[Rate on scale of 1 to 5 where 1 is strongly disagree and 5 is strongly agree]** It was more effective to learn through collaborative games than learning on my own in class (e.g. listening to lectures).
- **[Open-ended questions]**
  What did you like about the games in class?
  What didn't you like about the games in class? Please share ideas for improvements.

The observation notes were synthesized to draw out insights into the games' execution and outcomes. The games survey results were analysed and presented in graphical format by the Google Forms tools. The open-ended answers were manually analysed to identify the most significant concerns of the learners following thematic analysis (Braun and Clarke, 2006). The results of the surveys and qualitative findings are presented in the next section.

4 FINDINGS

Observations of the games played in class and the responses to the games survey showed a strong support for games-driven learning in classrooms in general and for learning about agile and lean software development concepts. We first present the observations made in the classroom, followed by the survey results pertaining to specific games played in the classroom and then report on learners' perceptions of the use of games for learning in general.

4.1 Classroom Observations

All learners were seen to engage in the Paper Planes game. High levels of teamwork and competition between teams was observed. In the first iteration, all teams over-estimated their productivity, most left testing to the end, running out of time. When the 'customer' (lecturer/author) inspected the deliverables (planes) at the end of the iteration, several planes were rejected because they did not meet the acceptance criteria. This led to teams paying closer attention to the customer's acceptance criteria in future iterations. In the 'check' (reflection) stage, teams could discuss what was working well for them and what improvements they needed to make; both to the design of the planes (technical improvements) and their approach to production and testing (process improvements). Most teams adjusted their estimates to more realistic numbers in the second iteration and commenced testing early in the 'do' stage, some ascribing a dedicated tester. The productivity rose incrementally until it peaked in the third iteration where teams had now begun to fine-tune their designs and process. The final debriefing session allowed learners to share their thoughts on the experience. Learners could appreciate the value of explicit 'plan' and 'check' stages in addition to the usual 'do' stage. This was particularly useful for the final year undergraduate students who are often seen to place more value on software coding or development and under-estimate the importance of design, planning, and reflection. They also observed the value of iterative delivery to deliver constant value to the customer and receive early feedback on design which could be incorporated into future iterations.

In the Name Game, one team member played the developer and the remaining members played the customers. Customers were seen waiting for their turn to interact with the developer. When asked about this in the debriefing session, most 'customers' agreed that they preferred longer waiting in some cases (e.g. the last few customers in queue) to receive the full, undivided attention of the developer to their request than to be served simultaneously alongside other customers. One of the customers didn't find any observable difference in their experience as they were the last to be served in both scenarios. In the real-world, however, with WIP limits in place, customers wouldn't necessarily need to wait idly while the developer is

serving others and can in fact attend to their own work until their request is ready for action.

It was seen that the whole class was engaged in the Planning Poker game, placing their hands up to represent their individual estimations by number of fingers and looking around to see with interest what their team mates and other peers estimated. The facilitator/author interacted with some teams to demonstrate the process of discussing and resolving estimation differences. It is important that the facilitator explains the importance of such discussions and clarifies that the intent is to enable outliers to voice their multiple perspectives rather than to pick on them.

Figure 1 shows some of the faces drawn as pairs during the Pair Draw game. The three drawings were drawn by three different learner pairs. Within each pair, individuals took turns to draw a single face feature using different coloured pens. The first drawing shows an approach where the two individuals drew different complete parts of the face (e.g. both eyebrows drawn in black by one and both eyes drawn in blue by another.)

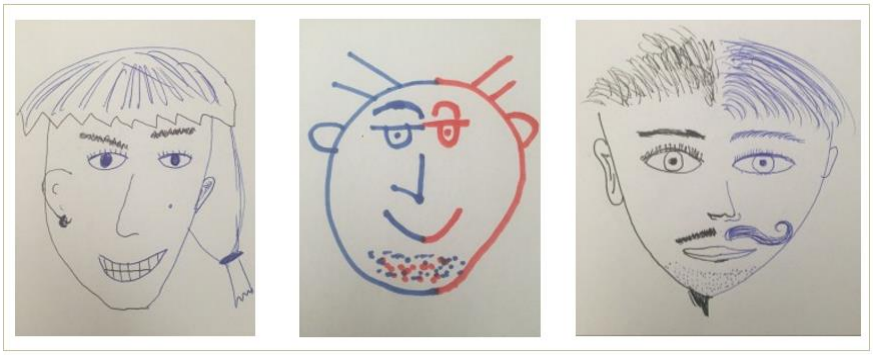

Fig. 1. Face drawings created in pairs during the Pair Draw game; each of the three drawings were created by a pair each using two different coloured pens (e.g. blue and black in the first drawing.)

The second and third drawings show an approach where the individuals attempted to mirror each other's actions on either side of the face. It was observed that often individuals tended to mirror their pair, e.g. if an individual drew the right eye, the other drew the left eye and so forth. Some were more creative in their efforts than others, but overall most pairs attempted to create a coherent face together. As the facilitator, it is important to reassure learners that the aim of the game is not to assess their drawing skills rather to allow them to experience working on a single task as pairs as some learners can be inhibited otherwise.

4.2 Games Survey Results: Specific Games

When asked about the ability of the game Paper Planes to teach the underlying agile concept of iterative delivery in the games survey, most learners (n=18 of 23) agreed or strongly agreed while only one learner disagreed.

We further probed about the game's effectiveness despite not involving any actual software delivery. Most of the respondents still agreed. Similar questions were asked for the remaining games.

Figure 2.A. Effectiveness of Planning Poker game to teach effort estimation. B. Effectiveness of Planning Poker game to teach about effort estimation, despite not involving actual user stories.

Figures 2.A, 3.A, and 4.A show the responses for the games Planning Poker, Pair Draw, and Name Game to teach estimation, pair programming, and work-in-progress limits respectively. Figures 2.B, 3.B, and 4.B show the same when emphasizing the non-software nature of the games.

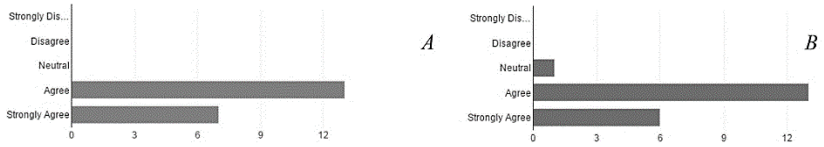

Figure 3.A. Effectiveness of Pair Draw game to teach about Pair Programming. B. Effectiveness of Pair Draw game to teach about Pair Programming, despite not involving actual programming.

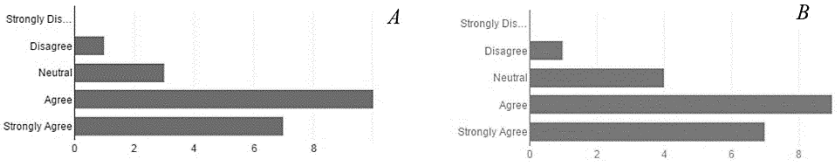

Figure 4.A. Effectiveness of Name Game to teach about Work-in-Progress Limits. B. Effectiveness of Name Game to teach about Work-in-Progress Limits, despite not involving software work.

4.3 Games Survey Results: General Questions

In introducing games into the curriculum, we wanted to include an optimum number of games in classroom sessions such that the learners felt neither over- nor under-whelmed. Most respondents said that the number was 'just right', while 3 learners said there were 'too few'. None of them thought that the number of games were 'too many' suggesting that we had managed to achieve a reasonable balance with four games in a three-weeks learning period.

We posed a series of questions aimed at assessing the learners' opinions of games for learning as a means to improve fun and enjoyment leading to better engagement in classroom learning. An overwhelming majority of learners enjoyed playing the collaborative games in class as seen from the results in Figure 5.A. Most of the learners (n=19 of 23) found collaborative games to be more fun than individual or solo learning in a classroom setting as depicted in Figure 5.B.

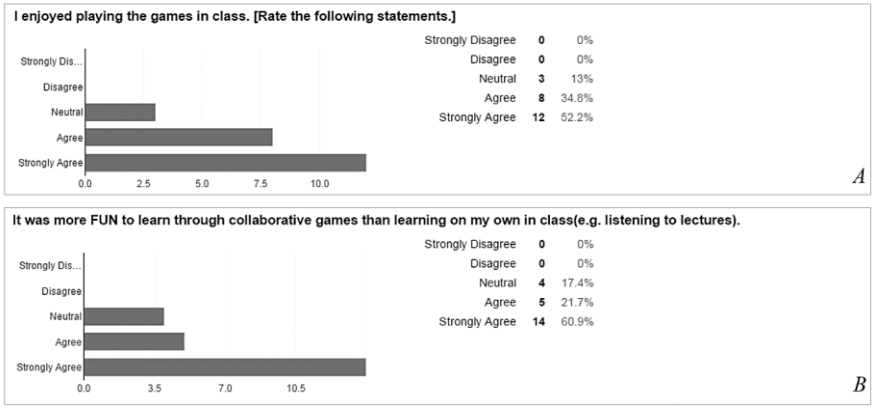

Figure 5.A. Perceived enjoyment in playing games in the class. B. Comparing collaborative games to individual learning in classroom based on perceived levels of fun and engagement.

While the ability of games to improve engagement is almost intuitive and well-established (Wangeheim et al., 2013; Scott et al., 2016), evidence to support their learning effectiveness can be elusive. We posed a series of questions to assess the learners' opinions of the games as effective classroom learning mechanisms. Most of the learners (n=20 of 23) agreed that they learned about the underlying agile and lean concepts effectively through the games played in class as shown in Figure 6.A. Furthermore, when comparing individual learning to collaborative games-based learning in classrooms, most (n=20) also agreed that the latter was more effective for their learning than the former, as depicted in Figure 6.B. Only one learner disagreed in each case.

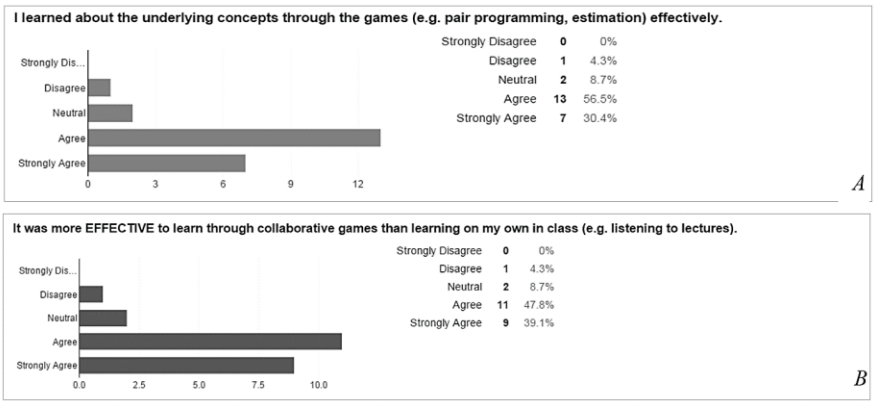

Figure 6.A. Perceived learning effectiveness of games. B. Comparing collaborative games to individual learning in classroom based on perceived learning effectiveness.

We also explored the learners' opinions on the ability of collaborative classroom-based games to assist with team building. An overwhelming majority of learners (n=22 of 23) agreed that collaborative games were good for team building, as shown in Figure 7.

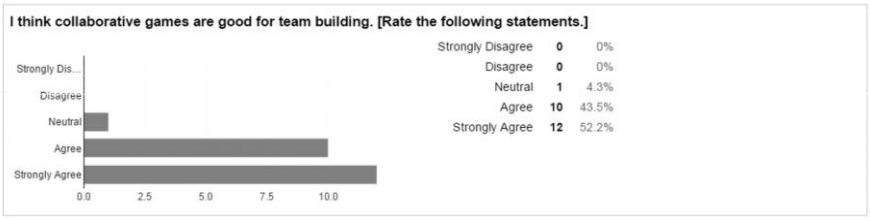

Figure 7. The ability of collaborative games to assist with team-building.

Finally, we asked a set of questions to gauge the learners' perceptions of games for learning in general. Many learners (n=16) agreed that teaching through games was both useful and necessary, as depicted in Figure 8.A. Some learners (n=6) found games to be useful for teaching, but not necessary and one learner responded by saying they neither found games useful nor necessary for teaching.

When asked about the appropriate role of games in delivery of content, most learners (n=19) selected games to supplement traditional delivery of content while a small percentage (n=4) said that games can replace traditional delivery, as shown in Figure 8.A and 8.B respectively. As seen in Figure 8.C, respondents (n=18) strongly/agreed that games should be used more often in classroom teaching in general, clearly implying a strong support for games-based learning in modern classrooms.

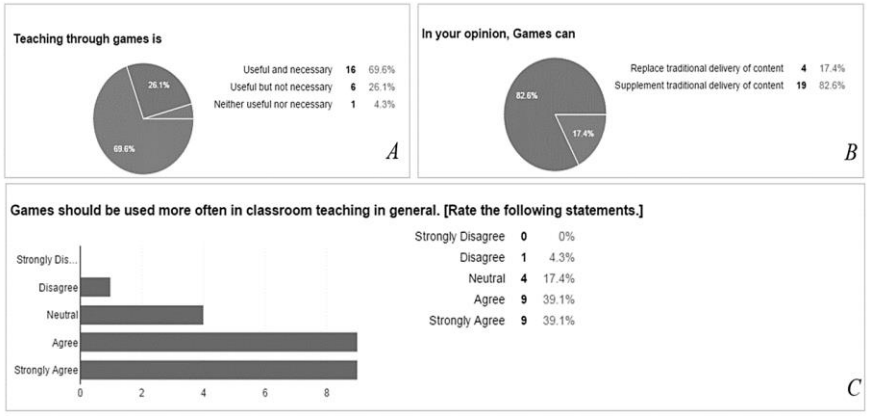

Figure 8.A. Teaching potential and role of games. B. The role of games in the delivery of content. C. The use of games in classrooms in general.

4.4 Games Survey Results: Open-Ended Questions

In addition to structured questions, we also included some open-ended questions. When asked about what they liked about the games in class. The responses to these questions were analyzed to group concept aspects together into themes. Here the main themes, italicized below, are presented along with relevant sample quotes.

Learners said that the *interactive nature of the games improved engagement* and saw their usage as *a welcome alternative to traditional lecturing*.

> "Helped to break up two-hour lectures keeping the class fresh and engaged."
> "It helped teach the concepts in Agile in a more fun and interactive way. Rather than listening to the lecturer go over the concepts, it was more fun to play a game and then understand how it relates to Agile."
> "It makes the learning more engaging."
> "It is fun. Classes filled by boring speeches make me sleepy. Games in class are really good for me to focus on the content easily. I can learn knowledge with pleasure."

It was also evident from the comments that some learners found games to be *an effective way of learning key agile and lean concepts*, for example:

> "Simple to do, which made it easier to understand the underlying concepts related the game."
> "Created a fun environment for pupils, understanding key concepts at the same time."

Many learners pointed to the collaborative games' ability to assist with *team building*, further strengthening similar observations from the classroom and the ratings questions:

> "All the team work together and we can get more closer."
> "It helped us work as a team. Our main aim was to succeed as a team rather than an individual."

When asked about what they didn't like about the games in class and ideas for improvement, two respondents noted that they didn't find Pair Draw particularly relevant to real world pair programming. The same observation resonated in the results of the ratings questions, where Pair Draw was rated least among the four games with regards to its ability to explain underlying agile and lean concepts. This may be because the Pair Draw exercise while presenting an opportunity to practice collaborative working, did not represent some of the other aspects of pair programming well, such as the ability to question each other, clarify uncertainties, make and correct mistakes, thereby carrying lesser real-world relevance than the other games.

Another improvement suggested by a few learners was *keeping the game instructions visible*. While instruction slides were provided for the Paper Planes and the Name Game games, other games such as Pair Draw and Planning Poker were described verbally.

> "A bit hard to know what exactly we had to. Maybe a slide or printout with the instructions instead of being mostly verbal."
> "The rules of the game are sometime confused. It is better to put the rules on slides and we can see it clearly."

We have since then added written instructions for these games in the classroom to cater to all types of learners, not just those who follow verbally. We also asked the learners about other agile and lean concepts they would have liked to learn through games. Some suggestions included: daily scrum, writing user stories, the pull system from Kanban, the role of the scrum master, and use of product and sprint backlog, and games that helped compare scrum, XP, and Kanban approaches. Of these, we have recently included the Kanban Pizza Game (Agile42, 2015) for simulating the pull system and WIP limit in Kanban.

5 DISCUSSION

Classroom observations and the qualitative and quantitative results of the games survey showed strong evidence in support of collaborative games in the classroom. The key findings, lessons learned and related implications for practice resulting from this study are discussed below and summarized in Table 2.

Table 2 Key Findings, Lessons Learned and Implications for Practice

Main Findings and Lessons Learned	Implications for Practice
The games effectively supplemented learning.	Educators and trainers should consider introducing collaborative games to supplement the content shared in traditional lecture-based delivery. Using them in isolation is not advised as typically some background information is required before learners can indulge in the games and gain insights from them. Games easily available online can be selected based on their ability to map to desired learning outcomes and graduate attributes.
The games improved classroom engagement.	Collaborative games can be used to easily invigorate the engagement of learners in the content and in the classroom in a fun, interactive way. Team-based games promote inter-personal interaction, attract interests from a wide variety of leaners, and tend to engage all learners instead of a selective few.
The games promoted team building.	Educators and trainers can use collaborative games to promote team building prior to commencing team-based agile projects.
Effective facilitation was vital.	Facilitators should not only explain the game rules and keep time but also actively encourage participation and discussion throughout the game. New facilitators should playtest with colleagues or assistants beforehand.
Debriefing was imperative.	Enough time should be allocated to debriefing each game. Learners should be encouraged to share experiences, insights, and questions. Most games include a list of questions to drive the debriefing session.

Based on the experience of introducing games in SOFTENG761 it can be said that games are a useful way to supplement the learning of agile and lean software development concepts and practices. The use of traditional lecturing using Power Point, in-class discussions and question-answers was seen to provide the necessary albeit brief background information before learners could engage in gameplay. With some background information, the learners were familiarized with the concepts underlying the game or exercise so they could draw better connections between the game experience and the learning content.

While most of the learners preferred games over traditional forms of delivery, most also subscribed to the view that games can supplement traditional delivery rather than replace it. Similar conclusions were drawn by Fernandes and Sousa (2010) and Wangeheim et al. (2013) who reported that their physical games PlayScrum and Scrumia respectively were best used in combination with traditional classroom instruction of the basic theoretical concepts. As such, trainers and educators, especially those in the academic domains, can consider selecting from a

plethora of Agile games available online. While not all may be suitable, games can be assessed and selected based on how well their various components map to the desired learning outcomes and graduate attributes (Table 1.)

Considered from a theoretical perspective, learning via games seems to embody reflection-in-action from Schön's theory of reflective practice (Schön, 1983) where reflection occurs during a given action (in this case, the game), as opposed to post the action occurring.

There was ample evidence of improved engagement from the classroom observations and the survey results. This is in line with similar assertions by Shneiderman (2004) and Paasivaara et al. (2014) who reported gains in learning and insight.

An additional finding from our study was that collaborative classroom games can be used effectively for team-building. While this was not the primary purpose of the games, they served as a good way to break the ice within the newly formed teams where members were not familiar with each other and promote cohesion in others. This can be especially useful in courses which include a practical project based component.

Effective facilitation was found to be useful. The facilitators not only explained the rules of the game and kept time for the different activities but also moved around the classroom to actively encourage participation and discuss how the different teams were executing the various game activities. It would be best for new facilitators to familiarize themselves with the game and playtest it with colleagues or teaching assistants prior to implementing in the classroom.

A dedicated debriefing session provided the opportunity to verbalize and establish the connections between the game experience and the underlying theoretical concepts. It allowed not only the learners to reflect on their game experience but also enabled the facilitator to discover new aspects about the game shared by learners. The importance of such post-game debriefing – embodying reflection-on-action (Schön, 1983) where reflection occurs after the action (in this case, the game) has finished – has been previously emphasized by Crookall (2010.)

5.1 Limitations and Future Work

While the facilitator and learners could play all the games within the given lecture-style classroom setup, it was observed that a different, more open-plan room setup would potentially be better for the game execution. For example, in the paper planes games, it would have been better to have an open area to test the airplanes. Similarly, for the name game and planning poker, a round-table seating for each team would have helped them face each other more easily and potentially improve collaboration. In the future, we aim to request for more open-plan rooms for our course.

The Pair Draw game received the least amount of support among all the four games. Many students had no prior concept or experience of pair programming and

linking the Pair Draw to pair programming may have been a wide leap for some. However, it is also a reflection on the relative low relevance of this game, Pair Draw, to the real Agile practice it was meant to represent, pair programming, as discussed earlier. We have since then dropped this game, and introduced the Kanban Pizza Game as discussed earlier.

Finally, while the evaluation of the games through student feedback, questionnaires and surveys is not the most robust or only method, it is a common method used in other studies on use of games for agile learning (Fernandes and Sousa, 2010; Lynch, 2011; Wangeheim et al., 2013.) The survey instrument used in this study was developed from scratch and some of the questions could have been better phrased for clarity and validity. For example, in hindsight, the question on the 'effectiveness' of the games for learning compared to learning on one's own could have captured responses comparing active and passive learning, rather than game and non-game based learning. To better address such issues in future studies, we plan to find, and if needed extend, validated and well-established questionnaires to assess learner satisfaction and explore the possibility of pre/post testing.

6 CONCLUSION

Games are widely used in industrial agile software development certification and training workshops. There is an increasing trend of trialing games for learning agile software development in academic contexts, in particular, the Scrum method and its practices. We introduced four paper-based collaborative games in a dedicated agile and lean software development course to help learners understand and experience concepts such as iterative and incremental delivery, team-based effort estimation, pair programming, and work-in-progress limits. Through classroom observations and survey-based quantitative and qualitative data, we found that collaborative games were preferred by most learners over tradition ways of individual learning in class, both in terms of their potential for fun and learning effectiveness. These collaborative games were found to supplement learning; helped invigorate learner engagement; and promoted team building. While new games originating from academic contexts are welcomed, educators can select from a variety of existing agile games easily accessible with resources online to introduce games into their classrooms based on their ability to deliver desired learning outcomes and graduate attributes. The successful use of the games, however, is dependent on effective facilitation by knowledgeable and experienced educators and on dedicated debriefing sessions that relate the game experience with the underlying theoretical concepts. Other types of games in academic contexts and industry-based professional agile certifications, trainings, and workshops can be studied in the future to gauge the impact of Agile games on engagement and learning effectiveness.

Acknowledgements: Sincere gratitude is due to the students of SOFTENG761 Agile and Lean Software Development course at the Department of Electrical and Computer Engineering at The University of Auckland, New Zealand.

References

Accord, W. (2013). Graduate attributes and professional competencies. Version, 3, 21. Available online: http://www.ieagreements.org/accords/washington/ (last accessed 8th May 2018)

Agile42, (2015) Kanban Pizza Game, https://www.agile42.com/en/training/kanban-pizza-game/ (last accessed 9th May 2018)

Anderson, D. (2010) Kanban: Successful Evolutionary Change for Your Technology Business, Blue Hole Press

Augustine, S. (2005) Managing Agile Projects. Prentice Hall PTR.

Baker, A., Navarro, E. & van der Hoek, A. (2005). An experimental card game for teaching software engineering, Journal of Systems and Software, 75(1-2): pp.3-16. DOI 10.1016/j.jss.2004.02.033.

Beck, K. (1999) Extreme Programming Explained: Embrace Change, first ed. Addison-Wesley Professional.

Braun, V. & Clarke, V. (2006) Using thematic analysis in psychology, Qualitative Research in Psychology, 3 (2), pp. 77-101, ISSN 1478-0887

Carrington, D., Baker, A., & van der Hoek, A. (2005). It's all in the game: teaching software process concepts, Proceedings of the 35th Frontiers in Education Conference.

Cohn, M. (2016) Planning Poker Game, Available at https://www.planningpoker.com/ (last accessed 9th May 2018)

Crookall, D. (2010). Serious Games, Debriefing, and Simulation/Gaming as a Discipline, Simulation & Gaming. Vol. 41. No. 6, pp. 898-920.

Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2010). The scrum primer. Scrum Primer is an indepth introduction to the theory and practice of Scrum, albeit primarily from a software development perspective, available at: http://assets. scrumtraininginstitute.com/downloads/1/scrumprimer121. pdf, 1285931497, 15.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. Software Development, 9(8), 28-35.

Hoda, R., Noble, J. & Marshall, S. (2013) Self-Organizing Roles on Agile Software Development Teams, IEEE Transactions on Software Engineering 39(3), pp. 422-444, IEEE.

Heintz, J. (2016) Agile Airplane Game, Gist Labs, Available at http://gistlabs.com/2011/06/agile-airplane-game/ (last accessed 9th May 2018)

Kerievsky, J. (2016) PairDraw, Industrial Logic, https://www.industriallogic.com/blog/pairdraw-2/ (last accessed 9th May 2018)

Kniberg, H. (2016) Multitasking Name Game, Crisp, https://www.crisp.se/gratis-material-ochguider/multitasking-name-game (last accessed 9th May 2018)

Kolb, D.A. (1984.) Experiential Learning: Experience as the Source of Learning and Development. Prentice Hall, Upper Saddle River, NJ.

Kruchten, P., (2011). Experience Teaching Software Project Management in Both Industrial and Academic Settings. In: Proc. of 24th IEEE-CS Conference on Software Engineering Education and Training, Honolulu/Hawaii, pp. 199–208.

Lindley, C.A. (2003). Game Taxonomies: A High Level Framework for Game Analysis and Design. Gamasutra Website: http://www.gamasutra.com/view/feature/131205/game_taxonomies_a_high_level_.php (last accessed 9th May 2018)

Lu, B. & DeClue, T. (2011). Teaching agile methodology in a software engineering capstone course. J.Comp.Sci.Coll. 26(5), pp. 293–299.

Mahnic,V. (2012). A Capstone Course on agile software development using scrum. IEEE Trans. Educ., 55(1), pp.99–106.
Malone, T. (1981). Toward a theory of intrinsically motivating instruction. Cognitive Science, 4, pp.333–369.
Paasivaara, M., Heikkila, V., Lassenius, C., & Toivola, T. (2014).Teaching students scrum using lego blocks. In: Proceedings of the 36th International Conference on Software Engineering, pp.382–391. URL:http://doi.acm.org/10.1145/2591062.2591169.
Percival, F., Ellington, H., & Race, P., (1993). Handbook of Educational Technology. Kogan Page, London.
Poppendieck, M. & Poppendieck, T. (2003). Lean Software Development: An Agile Toolkit, The Agile Software Development Series, Addison Wesley
Royce, W.W. (1970). Managing the development of large software systems. pp. 328-333.
Schön, D. (1983) The Reflective Practitioner: How Professionals Think in Action, Basic Books.
Schroeder, A., Klarl, A., Mayer, P. & Kroiß, C., (2012.) Teaching agile software development through lab courses. In: Proceedings of the Global Engineering Education Conference (EDUCON), IEEE. pp.1–10.
Schwaber, K. & Beedle, M. (2002). Agile Software Development with Scrum. Prentice-Hall.
Scott, E., Rodríguez, G., Soria, Á., & Campo, M. (2016) Towards better Scrum learning using learning styles, Journal of Systems and Software, Volume 111, pp. 242-253, http://dx.doi.org/10.1016/j.jss.2015.10.022.
Shneiderman, B. (2004). Designing for fun: how can we design user interfaces to be more fun? interactions 11(5), pp. 48-50.
Taiichi, O. (1988). Toyota Production System - beyond large-scale production. Productivity Press. pp. 25–28. ISBN 0-915299-14-3.
Wang, X., Conboy, K. & Cawley, O. (2012). ―Leagile Software Development: An Experience Report Analysis of the Application of Lean Approaches in Agile Software Development, The Journal of Systems and Software, 85(6), pp. 1287-1299.
Wangenheim, C. G., Savi, R., & Borgatto, A.F. (2013) SCRUMIA—An educational game for teaching SCRUM in computing courses, Journal of Systems and Software, Volume 86, Issue 10, pp. 26752687, ISSN 0164-1212, http://dx.doi.org/10.1016/j.jss.2013.05.030.
Wangenheim, C.G. & Shull, F.V. (2009). To game or not to game? IEEE Software 26 (2), pp.92–94.
Womack, J.P., Jones, D.T., & Roos, D. (1990) The Machine That Changed the World, Free Press, ISBN-13: 978-0-7432-9979-4
Wouters, P. et al. (2009) Current Practices in Serious Game Research: A Review from a Learning Outcomes Perspective, In M. Stansfield & L. Boyle (Eds.) Game-based Learning Advancements for Multi-Sensory Human Computer Interfaces.

Appendix: Survey Questions

Part 1 General Questions

- Number of games used in SOFTENG761 were: Too few / Just right / Too many
- Teaching through games is: Useful and necessary / Useful but not necessary / Neither useful nor necessary

- In your opinion, games can: Replace traditional delivery of content / Supplement traditional delivery of content
- Rate the following statements (Strongly Disagree/Disagree/Neutral/Agree/Strongly Agree)
    - I enjoyed playing the games in class.
    - I learned about the underlying concepts through the games (e.g. pair programming, estimation) effectively.
    - Learning through playing games teaches concepts in a way that traditional lectures cannot.
    - It was awkward to play games in class.
    - Games cannot teach concepts effectively.
    - It was more EFFECTIVE to learn through collaborative games than learning on my own in class (e.g. listening to lectures).
    - It was more FUN to learn through collaborative games than learning on my own in class (e.g. listening to lectures).
    - I think collaborative games are good for team building. o Games should be used more often in classroom teaching in general.
- Open-ended Questions
    - What did you like about the games in class?
    - What didn't you like about the games in class? Please share any ideas for improvements.
    - What other Agile and Lean concepts would you have liked to learn through a game-based learning approach?
    - What other kinds of learning approaches or content delivery styles (other than games) would you like us to try in SOFTENG761?
    - Would you like to play such games online to learn other Agile and Lean concepts?

Part 2: Game-specific questions (repeated for each game, shown for Pair Draw here.)

- Did you play PAIR DRAW in the class? Yes / No
- Rate the following statements (Strongly Disagree/Disagree/Neutral/Agree/Strongly Agree)
    - I found the game effective in teaching me the concept of PAIR PROGRAMMING.
    - I am more likely to attempt PAIR PROGRAMMING as a result of playing PAIRDRAW.
    - PAIRDRAW helped me understand the different aspects of PAIR PROGRAMMING even though it did not involve actual programming.

- I expect to have similar experiences with PAIR PROGRAMMING on the actual project as I did with the PAIRDRAW simulation in class.