

Respond to Change or Die: An Educational Scrum Simulation for Distributed Teams

Emily Laue Christensen
emic@itu.dk
IT University of Copenhagen
Copenhagen, Denmark

Maria Paasivaara
maria.paasivaara@lut.fi
LUT University
Lahti, Finland

ABSTRACT

Scrum is best learned by doing, e.g. through simulation, as it is simple to understand but difficult to master. During the Covid-19 pandemic, we could not use the traditional face-to-face Scrum Lego simulation game, but had to utilize something workable in an online environment. In this paper we present an online Scrum simulation for distributed teams that was created in a multiplayer game "Don't Starve Together" (DST) through iterative reflective work and analysis. We ran the simulation with 25 Scrum teams on four different courses with participants from eight universities located in three countries. We collected feedback from 244 participants by analysing 196 student learning diaries and 84 student evaluation surveys, and by running a retrospective with 19 industry participants. The participants' feedback was highly positive. The main reported learning outcomes were communication, estimation, Scrum in practice, communication and collaboration with industrial Product Owners, Scrum events, work organisation, teamwork and prioritisation.

KEYWORDS

software engineering education, Scrum, simulation, play

ACM Reference Format:

Emily Laue Christensen and Maria Paasivaara. 2022. Respond to Change or Die: An Educational Scrum Simulation for Distributed Teams. In *44th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '22)*, May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3510456.3514145>

1 INTRODUCTION

Agile software development, in particular Scrum [39], has become highly popular in industry. According to the latest State of Agile report [41], 81% of the respondents are using Scrum or a Scrum-based framework, such as Scrumban. Thus, there is clearly a need to teach Scrum to university software engineering (SE) students. Scrum is based on face-to-face conversation. The sixth principle behind the agile manifesto states that face-to-face conversation is "the most efficient and effective method of conveying information to and within a development team" [4]. In spite of that, agile, especially

Scrum, has been used successfully in global software development for over a decade, as it creates a structured way to communicate frequently for distributed teams [29, 35].

The Covid-19 pandemic forced most software developers to work from their home offices, which reshaped the ways software companies work [12, 18]. According to the recent studies during the pandemic [12, 18, 31], fully virtual collaboration has worked surprisingly well and many software developers feel that their productivity has stayed almost the same [31]. As the move to virtual work was smooth for many and even created benefits to software developers' personal lives, such as spending less time commuting and more time with the family, many of them are not planning to return to office full-time in the post-pandemic world. Instead, according to the State of Agile report, most respondents plan to continue working either in hybrid (56% of the respondents) or in fully remote mode (25% of the respondents) [41]. It seems clear that distributed software development is here to stay. Therefore, we need to train our students to be able to use agile and Scrum also in distributed circumstances.

The pandemic forced us to find teaching tools for this new reality, as we had to quickly learn how to best teach student teams Scrum online. According to the Scrum creators, "Scrum is simple to understand but difficult to master" and can only be learned "by doing" [43]. Therefore, it is important that students can try out Scrum in practice and learn by doing. The world is gaining complexity and effort that is put into knowing the unknowable and trying to restrict change is wasted [43]. Students need to learn how to respond to change [3], or as stated by Sutherland and Sutherland, "change or die" [43]. Previously, we had used the Scrum Lego simulation game to teach Scrum in practice (see, e.g. [5, 8, 16, 22, 27, 34, 42]), however, using a physical face-to-face simulation was not possible during the Covid-19 lockdown. Instead, we had to use something new that would suit the online learning environment. For this purpose we utilized an educational Scrum simulation that was designed and created by the first author in February 2020, prior to the pandemic, for a team of students who were collaborating online. The simulation worked surprisingly well both as a tool to practice Scrum and as a team-building activity. Therefore, it was further developed and in this paper we want to share our experiences and results on 1) how the Scrum simulation for distributed teams works, and 2) what the student learning outcomes are.

The Scrum simulation was designed through a Critical Technical Practice (CTP) [1], which aims to identify dominant discourses and normative trends in the field of SE education and test out alternatives [25], through digital play design [40]. According to Sicart [40], play is a way of being in and making sense of the world, through objects towards others. Play lets us understand the world

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICSE-SEET '22, May 21–29, 2022, Pittsburgh, PA, USA

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9225-9/22/05...\$15.00

<https://doi.org/10.1145/3510456.3514145>

Table 1: Games used for teaching agile and Scrum

Game	Year	Format	Intended Learning Outcomes
SimSE [33]	2004	Graphical computer game	Work in teams with large scale projects, critical decision-making, handling personnel issues and multiple stakeholders, budgets and planning, and dealing with unexpected events
Scrum from Hell [6]	2004	Card game	To practice standup meetings with a dysfunctional one
The Scrum Game [47]	2007	Board game	The Scrum lifecycle; the various stages in Scrum and the manner in which Scrum roles interact with each other
The Scrum Ball Point Game [7]	2008	Game with balls	Basic agile principles and values; trust, self-organization, adapt and inspect, timeboxed, incremental delivery
PlayScrum [17]	2010	Card game	Scrum concepts and practices, and associated practical issues
Scrum Game: An Agile Software Management Game [20]	2011	Prototype of a system in the form of a game	To guide students through the Scrum lifecycle, the different phases of Scrum, the way that the different Scrum roles interact with each other, and the way that Scrum is used to implement real projects
SCRUMIA [45]	2013	Paper and pencil game	Scrum roles, events, and artifacts
The Lego Scrum Simulation Game [34]	2014	Game with Lego blocks	Scrum process and roles, requirements management and customer collaboration, estimation, working in a team, and visualizing work and progress
SCRUM-X [28]	2016	Computer-based simulation game	To strengthen understanding of Scrum
Scrum Paper City [21]	2017	Multi-week paper game	Agile collaboration and values, Scrum basics; roles, events and artifacts, timeboxing, and estimation
SCRUMI [13]	2017	Electronic board serious game	Key Scrum concepts; early stage preparation, Sprint Planning, division of tasks, updating a task board, burndown charts, handling unforeseen events, monitoring and control, Sprint Reviews and Sprint Retrospectives

and through that understanding we can challenge the establishment and create new ties, or break old ones [40].

The created Scrum simulation is carried out within an online multiplayer game, “Don’t Starve Together” [26]. A prototype of the simulation was tested in March 2020 by one distributed Scrum team and after small modifications, we ran the simulation a total of 25 times, with 244 participants of which 220 were BSc and MSc students, in four different courses, with eight different universities. All courses were organised online. The first two courses were project-based. In these courses the Scrum simulation was also a team-building activity for the participants that included, besides the students, also their collaborators from industry: Product Owners (PO) and agile coaches. After the simulation these same Scrum teams run a three month software project. The last two courses were primarily lecture-based, without a software development project. The teams ranged from 4-11 students, with an average of 9 students in each team. We collected data from the participants on their simulation experience and learning. We analysed 196 student learning diaries and 84 student survey responses, as well as organised a retrospective meeting with nine POs and ten agile coaches who had participated in the simulation.

In Section 2, we present the background for this paper, including the related work, digital play design, and the choice of the simulation format and environment. In Section 3, we describe the Scrum simulation. In Section 4, we describe the research method. We present the results in Section 5, and conclude the paper in Section 6.

2 BACKGROUND

2.1 Related Work

According to Cico et al. [11], the main approaches for teaching SE include project based learning, gamified learning, blended learning, and experiential learning. In this work, gamification and the dominant methods for teaching agile and Scrum, within the gamified learning approach, have been further explored, with the aim of identifying alternative teaching methods. Our literature review revealed that within the gamified learning approach, games have primarily been used to teach agile and Scrum in SE education. The format and intended learning outcomes for these games are listed in Table 1. As can be seen from the table, a few of the games are computer-based, however, none of them are online games, nor do they allow for the whole team to participate together, but are used individually by students. Therefore, they did not fit to our purposes, as we expected the Scrum simulation to have also a team-building aspect. There exist also several different versions of the Lego Scrum simulation game [5, 8, 16, 22, 27, 34, 42]. During a typical Lego Scrum simulation game students act out their Scrum roles, which includes developers and the Scrum Master (SM), and the leader of the game is often a teacher who acts as a stereotypical PO [34]. The practice of teachers and internal university staff acting as clients in SE education is also discussed by Cico et al. [11], who find that this is often practised in course settings when an industry role is missing. This simulation of roles is, however, only partially effective, as it does not provide any real external pressure [11]. The students have reported that their main areas of learning from one reported version of the Scrum Lego simulation game were: Scrum process and roles, requirements management and customer collaboration, estimation, working in a team, and visualising work and progress [34]. For the online simulation, we aimed to have all these learning

outcomes of the Scrum Lego simulation game, as well as include the real industry PO.

2.2 Digital Play Design

2.2.1 Playful simulations. In contrast to gamification and games, playful design “implies a system, which has playful elements in its design, but also has components which are not playful and the system has a non-playful, real-life, purpose” [24]. Play and simulation have the potential to generate discovery and the integration, application and sharing of knowledge, within collaborative learning environments [9]. Play and simulation require that “students learn to make and trust their own judgements” [9], and can therefore be used as effective ways of generating learning.

2.2.2 Digital play spaces. According to Sicart [40] play both modifies, and is modified by, physical or virtual environments. In contrast to a game space, which is specifically designed for a game activity, a play space is a location that accommodates play, but does not impose any particular type of play, set of activities, purpose, goal, or reward structure [40]. Playgrounds are useful metaphors for this differentiation between concepts and can allow us to escape from game spaces, which are designed for the purpose of playing games, but do not always allow the exploration of the creative and appropriative capacities of play. While most virtual game worlds are designed to support a particular game, sandbox games in which players can more or less freely roam an expansive virtual environment can be considered both as game spaces and play spaces. These open-world sandbox games can therefore be described as digital playgrounds, or spaces designed for appropriation, yet populated by props that help steer predetermined activities [40]. Chang argues that open-world video games can be viewed as “richly designed problem spaces” or “possibility spaces” [10], where players can be confronted by our current environmental knowledge and impact. Environmentally realistic games can enhance understanding of real-world environmental issues, either by implicitly or explicitly modeling different forms of individual and collective environmental agency [10].

There are however strong norms, rules and laws that govern how people use public and private spaces and play design must be carried out accordingly. Sicart argues that these norms and regulations are often conservative estimates based on the types of play that are deemed correct and those are often “based on fear rather than on the potential for play to be an expressive way of being in the world” [40]. This notion is reflected in the increase of playgrounds built for safety, rather than for play, in accordance with protective laws. An example here are the certified safe Monstrum playgrounds, which are the latest iteration of a Danish invention, “the adventure playground”. These were spaces originally created by Danish pedagogues who wanted to provide children with the tools to build their own playgrounds, thus letting them express themselves through play. A careful balance is however needed when exposing children to the creative and potentially destructive capacities of play, and the obvious dangers of providing children with tools, like hammers and nails, resulted in a ripple effect; safety was moderately ensured through adult supervision, but this also meant that the play was monitored and potentially interfered with [40]. This example demonstrates the normative idea that play is



Figure 1: Don't Starve Together

more secure when controlled, which is arguably also reflected in virtual play spaces.

2.3 Choosing the Format and Environment

Based on the above findings the first author designed and created a playful digital simulation that could be used by student Scrum teams, including their industry POs. This simulation should both facilitate learning about Scrum and communication in a distributed setting, and act as a team-building activity. For the Scrum simulation environment, or playground, an online game was chosen: Don't Starve Together (DST) [26]. This choice ties into the theories by Sicart [40] and Chang [10]: DST is an ideal digital playground, because it is an environmentally realistic open-world sandbox game [10]. The characters in DST are also very diverse in terms of gender and abilities, which is a realistic representation of individuals in cross-functional teams. Staying alive in DST requires constant and careful maintenance and there are no tutorials, rules, points, or specific objectives [23]. By using DST as the digital playground for the simulation and embracing the dangerous, risky, and disruptive qualities of this open-world play space, an alternative to the normative idea that play is more secure when controlled could be tested.

3 DST SCRUM SIMULATION

The Scrum simulation is primarily carried out in the DST open-world game environment (Figure 1), therefore each participant must own a copy of the game. During the simulation communication is facilitated through voice channels in Discord [14], an instant messaging platform. Participants can also text and share media files in a designated text channel, e.g. screen shots of the game map. The agile board used for the simulation is screen shared in Discord during specific events. The board is created in Trello [2], a web-based, list-making application.

3.1 Learning Goals

The main goal of the DST Scrum simulation is to provide a digital play space where participants can learn Scrum in practice, while simultaneously providing an opportunity for team-building. The learning goals are the following:

- (1) Communication within the student team and with an industry PO
- (2) Scrum events, roles, and artifacts
- (3) Agile values and principles
- (4) User stories, prioritization, and estimation
- (5) Time management
- (6) Teamwork and team-building

3.2 Don’t Starve Together

The original single player version of DST, Don’t Starve, was marketed as an uncompromising wilderness survival game full of science and magic, but is described by one of the creators as “a survival, horror, crafting, death simulator” [19]. The main character is a scientist named Wilson who is trapped in a mysterious wilderness. Players must guide Wilson to explore his world, use the affordances provided, and avoid death. The multiplayer expansion, DST, can be purchased on Steam for 14.99€, with two copies of the game included in the price. When players enter the game world, the map is completely black and the whole world must be explored before all areas can be viewed. Game worlds consist of a combination of biomes, which are usually differentiated by the type of terrain they are composed of [46]. There are 17 different biomes on the surface of the world in DST and 13 underground. Players must collect the various resources, such as grass, twigs, flint, logs, rocks and gold, to ensure survival and build items. As one of the most important items in the game, food is typically very valuable. The availability will vary depending on the world generation and layout. Food can be collected by foraging, trapping, farming, or hunting.

DST also has four different seasons in which the environment changes, which in turn changes the game-play [46]. The game begins in autumn by default and progresses through winter, spring, and summer, before autumn begins again. Food abundance is affected by the seasons, being more available in autumn and spring. One in-game day in DST lasts for eight real-time minutes that can be viewed in a day counter on the player’s screen, which is split into sixteen 30-second segments (see Figure 3). After each day, the counter will increase. Each day consists of three phases; day, dusk and night. Day phase is the safest phase and also the time when most animals are active. Most monsters, like spiders, sleep during the day. During dusk the world becomes slightly darker and most animals will return to their homes. Monsters like spiders and mosquitoes become active during this phase and begin to roam the areas close to their homes. When night falls the world becomes completely dark and a light source is necessary to see, and to interact with objects, as well as to avoid attacks from the monster Charlie.

Character well-being is primarily measured by three factors in DST; hunger, sanity, and health. Hunger drops automatically over time and can be regained by eating food. Sanity drops automatically during dusk and night and will also be lost if the player performs certain actions, e.g., eating raw or spoiled meat. Players’ health will drop if they are e.g. attacked, injured, or starving. Health can be regained through e.g. sleep, eating certain food, or using specific items. When a players’ health reaches zero they will die and drop all their items. If playing in endless mode they will become ghost players and can be revived, e.g., by haunting the spawn portal, or



Figure 2: Product Goal

by another player. All of these game mechanics combined ensure that staying alive in DST is challenging in its own rite; “It pits an emergent idyllic beauty against a grim reality of unsustainability and entropy” [37]. The game also forces players to learn optimal play, while continuously reminding the player about the vibrant wonder and beauty they are destroying [37]. One example here are the narrative descriptions displayed when capturing and killing animals; the action to kill a rabbit is “murder”, and to place a bird in a birdcage it must be “imprisoned”.

There are currently 14 different free characters in DST [46]. Each character has its own unique perks and disadvantages, in addition to varying bonuses and penalties to their hunger, sanity, and health stats. This ensures a wide range of different play styles for DST players. The variety of the characters fully represents the variety desired in a cross-functional team, as each character has its own unique skills, weaknesses, and identity. The gender differences in the characters are also reflective of the actual world, including the inability to determine the gender of some characters, without referring to their gender pronouns. According to Hook [32], men identify equally strongly with fictional characters of both genders, but women identify more strongly with female characters, than male characters. This in mind, DST provides an inclusive play space, as players can pick their own character based on their personal preferences, which could contribute to a stronger identification with the character.

3.3 Simulation Design

For the in-game simulation activities a preconfigured dedicated server was created in DST which allows a maximum of 16 players. The server is set on endless mode and is slightly modified to provide a few more resources, and less hostile monsters. Participants must try to fulfil the Product Goal [39], by building a base for the PO.

3.3.1 Handout. The Product Goal is described in the handout¹ that participants receive prior to the simulation, through five epics: 1) food, 2) light, 3) a place to sleep, 4) companionship, and 5) safety (Figure 2). A Definition of Done (DoD) is predefined in the handout and it requires that materials are stored away and items have been tested by the PO. Facilities must also be safely integrated, meaning they are far enough away from each other to avoid fire spreading.

¹Handout: <https://doi.org/10.6084/m9.figshare.16850632.v1>



Figure 3: Time Frame



Figure 4: Different Pen Sizes

Fire is the most dangerous element in DST as it can quickly destroy completed work, as well as entire worlds. The handout also contains an introduction to DST, the simulation objectives and flow, the backlog work, the time frame, an overview of the agile board, the release planning activities, Scrum roles and events, and links to useful mods for DST.

3.3.2 Timeframe and Flow. The time frame for the simulation is set at 3.5 hours, which allows for 15 minutes of buffer time (Figure 3). During the first 20 minutes participants join the DST server and are provided with a few reminders about the simulation flow, Product Goal and DoD by the facilitator. The team then has 25 minutes to carry out an initial release planning and 40 minutes to complete the first Sprint. After a 15 minute break, two more Sprints are carried out. Finally, the team spends 15 minutes reflecting on the activity. During the simulation participants carry out four Scrum events: Sprints, Sprint Planning meetings, Sprint Reviews, and Sprint Retrospectives. Sprint Planning is timeboxed at eight minutes, while the Sprint Reviews and Retrospectives are timeboxed at four minutes each. The development work is timeboxed at 24 minutes, corresponding to three in-game DST days. Daily Scrums are not included in the time frame, but a number of teams held them during the night at their own initiative, as participants often return to the base during this time. The development work and the Sprint Reviews are carried out in-game, while the rest of the time is spent in discussion through Discord while viewing the agile board. Participants are instructed to pick their characters after the first Sprint Planning, prior to entering the DST game world. The choice of characters is left to participants, but should ideally be discussed within the team, as it is beneficial if there is a wide variety.

3.3.3 Agile Board and Product Backlog. During release planning and Sprint Planning, participants can view the agile board which is used for the simulation, as it is screen shared by the SM or PO. The board that was used for the first prototype and the two first courses consisted of four lists: Product Backlog, Sprint Backlog, Ready for Review, and Done. In the last courses two more lists for the Sprint Goals and Retrospectives were added, to increase transparency during the simulation. In the Product Backlog list there are a large number of cards or Product Backlog items (PBI). The remaining lists are empty at the start of the simulation. The PBIs consist primarily

of items needed to build a basic base and survive in DST, but the list is purposely quite long (around 30 items), as the idea is to simulate an endless list the PO may want, which should help the team learn the importance of prioritisation. Nothing in the Product Backlog is however set in stone and participants are encouraged to split, delete, or create any cards they like.

All of the PBIs have titles which correspond to items, tasks, animals, or animal pens in DST. The PBIs are also marked with labels to show if they are a user story or a task and which epic story they fall under. Each user story contains a description in the card following the template “As (the PO) I would like (item or animal), so I (receive benefit)”. E.g.; “As the PO I would like a tent to sleep in during dusk and night, so I can generate health and sanity”. Under the description there is a checklist with the steps that must be followed to complete the story. The first step in most user stories is to carry out research. This can be performed either by searching for the item on the DST Fandom Wiki [46], or by standing next to a science machine or an alchemy engine in-game and viewing the item under the respective tab. The next steps include collecting materials, refining materials, crafting the item, and testing. The checklists for the three cards containing pens for different animals follow the same format, but are slightly more ambiguous, as there are no specific instructions about the required size of the pen. The pens can be built in a variety of sizes (Figure 4), so the specific requirements should ideally be discussed among participants before the development work begins. This is one scenario that participants often struggle with, as it is quite difficult to lure or place the animals in the pens, if they are built very small.

The checklists for the animal cards are more descriptive and individualised, as each animal must be captured or lured in different ways. E.g., to catch a rabbit participants must: place a trap close to a rabbit during daytime, put bait in the trap, and then wait for the rabbit to be caught in the trap. There are also multiple dependencies between the PBIs, which will be discovered if participants read the user stories and tasks. E.g., to catch a rabbit, the trap card must first be completed. These dependencies are often overlooked by participants who do not read the contents of the cards. The PBIs labeled as tasks contain only a checklist with the various materials that must be collected. The checklists are slightly ambiguous, as they do not specify the number of materials that must be collected. This will vary, depending on which of the other PBIs have been

prioritized and should be discussed by participants. Some of the items may also be irrelevant depending on the PBI prioritization, and said items should ideally be removed or deleted from the card by participants.

3.4 Roles

In the first two courses the students carried out their roles as developers and SM, and the role of the PO was carried out by the industry partner. The industry agile coaches also joined the simulations in the first two courses and assisted by coaching their teams in Scrum and agile. The PO role was carried out by a student in the third course and by the teacher in the fourth course, and the SM role was carried out by the facilitator in the last two courses.

3.5 Facilitation

All of the simulations, with the exception of the second course, were facilitated by the first author. The facilitator was primarily responsible for hosting the DST Scrum simulation dedicated server, ensuring participants quality of life in-game, maintaining the timeboxes for Scrum events, and coaching the teams in Scrum and estimation. Maintaining the timeboxes is particularly important during facilitation, as survival in DST becomes more difficult as time progresses, which may impede the teamwork and learning. In the second course the simulations were facilitated by the course teaching assistants (TA), after they received training from the first author. The TAs were first provided with the dedicated server files² and instructions for installation³. The training consisted of an introduction to the basic game mechanics, followed by a 3.5 hour walk-through of the Scrum simulation where the first author facilitated and the TAs carried out the Scrum team roles. The TAs were also provided with a facilitation guide which included a list of the DST command codes and in-game items they would need to use⁴. All of the facilitators played in “god-mode”, which ensured that their characters could not die during the simulation.

4 RESEARCH METHOD

The method Critical Making [38], which combines critical thinking with iterative goal-based material work, was applied in our study. According to Ratto [38], a disconnect exists between our material connection to technology and our conceptual understanding of it. By focusing on a constructive practice in our work, we can address this disconnect and question how computational objects are made, while simultaneously exploring the aspects of materiality and relations that take place, when making things. Critical Making consists of three stages. In the first stage researchers should carry out a literature review of the topic being explored. Following this, development work on a prototype should be undertaken, to explore the possibilities of the design. Finally, an iterative process of “wrestling with material prototypes” [38] should take place, with the role of the materiality of the design being examined in each iteration. These stages were carried out by the first author over the duration of one year. The feedback was collected through a mixed-method approach, combining qualitative and quantitative methods.

²Dedicated server files: <https://doi.org/10.6084/m9.figshare.19164200.v1>

³Dedicated server installation guide: <https://doi.org/10.6084/m9.figshare.19164173.v1>

⁴Facilitation guide: <https://doi.org/10.6084/m9.figshare.19164176.v1>

Table 2: Data collection and analysis

Data source	Course	Collected data	Analysis
Evaluation surveys	1 (Oct 2020)	24 answers	Quantitative analysis
	2 (Feb 2021)	21 answers	
	3 (Mar 2021)	27 answers	
	4 (Apr 2021)	12 answers	
Learning diaries	1 (Oct 2020)	26 diaries	Qualitative coding
	2 (Feb 2021)	139 diaries	
	3 (Mar 2021)	31 diaries	
Retrospective	2 (Feb 2021)	9 Product Owners	Qualitative analysis
		10 Agile coaches	

The results from the data collection and analysis were taken into consideration and implemented in the simulation design following each course, ensuring continuous improvement. A summary of the data collection and analysis methods is shown in Table 2.

4.1 Research Questions

In this paper we aim to answer the following questions:

RQ1: What were the main learning outcomes of the DST Scrum simulation, as reported by the students?

RQ2: What were the experiences and perceptions of participants on the DST Scrum simulation?

RQ3: How could the DST Scrum simulation be further improved to enhance learning opportunities?

4.2 Data Collection

4.2.1 Learning Diaries. 196 learning diaries were analysed using qualitative coding [30]. The students from the first three courses were required to write a learning diary of 1-2 pages on their experience with the DST Scrum simulation. In the diary they were asked to discuss the main points learned during the simulation, reflect on the learning experience as a whole, and critically discuss and compare their learnings from the simulation with what they had learned in earlier lectures, course materials, or previous experiences in software development. They were also asked to discuss the simulation as a format of learning, e.g. by comparing it with other possible ways of learning Scrum and agile. Finally, they were given an option to provide feedback on the simulation and suggestions for improvement. All students were provided with a template for the learning diary which contained instructions for the structure and use of headings and sub-headings⁵. Two examples of subheadings were included in the template, which may have influenced the diary contents: 1) communication with the PO, and 2) story points and estimates. For the students in the third course, the first subheading example was changed to “communication” in the template. All students were informed that their learning diaries would be analysed as part of this data collection and they had the opportunity to opt out, but none requested this. The learning diary data is personal, thus we cannot make that data publicly available.

4.2.2 Surveys. All students were asked to fill out a survey about the simulation experience and their learning, which was completed by 84 students⁶. The survey was created in Limesurvey and the five

⁵Learning diary template: <https://doi.org/10.6084/m9.figshare.16850596.v1>

⁶Simulation survey: <https://doi.org/10.6084/m9.figshare.16850569.v1>

Table 3: Themes and codes (# of student diaries discussing the topic)

Learning goals	Other points learned	Experiences and perceptions
Communication (151)	Work organization (79)	Complexity (33)
Estimation (124)	Problem solving (35)	Team-building (29)
Scrum in practice (122)	Ask for help (23)	Timeboxing (28)
Communication and collaboration with PO (92)	Self-management (23)	Scrum roles (25)
Scrum events (81)	Work in pairs (20)	Simulation format (23)
Teamwork (66)	Inclusivity (18)	
Prioritization (52)	Research (16)	
Scrum roles (37)	Handling failure (14)	
Scrum artifacts (29)	Domain knowledge (14)	
Time management (28)	Knowledge sharing (13)	
Agile values and principles (15)	Dependencies (13)	
User stories (9)	Work breakdown structure (12)	
	Scrum values (12)	
	Trust (11)	
	Distributed work (9)	
	Transparency (8)	
	Pair programming (8)	

main question groups were modified slightly for each course. In the first question group the students were requested to evaluate their learning during the simulation on topics related to Scrum and agile. In the following question groups students were requested to evaluate statements on a five point Likert scale, ranging from strongly disagree to strongly agree, related to their motivation, user experience, and learning. In this part we used questions from a publicly available survey instrument that had been created to evaluate another educational game [44]. The last question group was only used in the first two courses. Here the students were requested to consider their own software project team in the course and evaluate the impact of the simulation on their project work. Finally, all students had an option to write their thoughts regarding the simulation, or add suggestions on how to improve the simulation in the future. We made the survey data available⁷.

4.2.3 Retrospective. A one-hour retrospective was carried out with nine POs and ten agile coaches in the second course, facilitated through a Miro board. Participants provided their feedback on what was good and what could be improved before, during, and after the simulation. This was followed by an open discussion between participants where notes were taken to collect the feedback. The feedback from this retrospective was incorporated into the simulation design and is presented in Section 5, along with other results.

4.3 Data Analysis

After each course an initial familiarisation with the data was conducted to get a thorough overview, then all the diaries were read, the main student learning outcomes were manually highlighted and the first set of codes was created. These codes were improved on by adding additional codes during the coding process and defining them. In the second course we learned that a few teams had experienced technical issues in the simulations and noted several perceptions about the simulation format and the application of agile and Scrum in practice. The code generation was then expanded

⁷Survey data: <https://doi.org/10.6084/m9.figshare.16863241.v1>

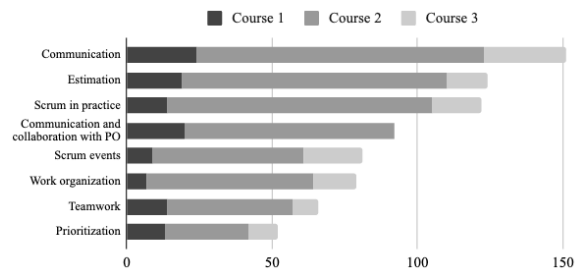


Figure 5: Learning diary results

to include these experiences and perceptions. An overview of the themes and codes can be viewed in Table 3. The entire coding process was carried out via computer and the results of the analysis were extracted to a data sheet in Excel, and summarised in graphs.

5 RESULTS

In this section, we present answers to our three research questions. Each question is answered in a separate section.

5.1 Main Learning Outcomes

In this section, we answer RQ1: “What were the main learning outcomes of the DST Scrum simulation, as reported by the students?”

5.1.1 Learning Diary Results. In this section we have compiled the main points learned by the students in the first three courses, as reported in the 196 DST Scrum simulation learning diaries. The five points reported the most often, through explicit descriptions and reflections, were 1) communication, 2) estimation, 3) Scrum in practice, 4) communication and collaboration with the PO specifically, and 5) Scrum events (Figure 5).

Communication was reported as one of the main points learned by 77% (151/196) of the students, as the following quotes demonstrate:

“DST Scrum simulation taught me that communication between team members is more important than recording sprint tasks in your head and focusing solely on completing it.” — Student developer A, Course 1

“Once communication had improved within the team, we all worked better together, because no one was afraid of admitting if they were confused. This experience taught me that it is crucial to have a safe-to-fail environment when working together as a team.”

— Student developer B, Course 2

Estimation was reported as one of the main points learned by 63% (124/196), followed closely by learning how to use Scrum in practice, which was reported by 62% (122/196). In the learning diaries from the second course in particular, we could see that many students expected estimation to be very easy and had not understood why this topic had been focused on during the course lectures. During the simulations their understanding of the topic deepened, especially in regard to estimation of tasks where the requirements were unknown or there were elements of uncertainty, and they learned the value of applying this agile technique:

“I found that in practice, problems can arise, and especially estimating the size of some tasks can be very difficult, when I had never played the

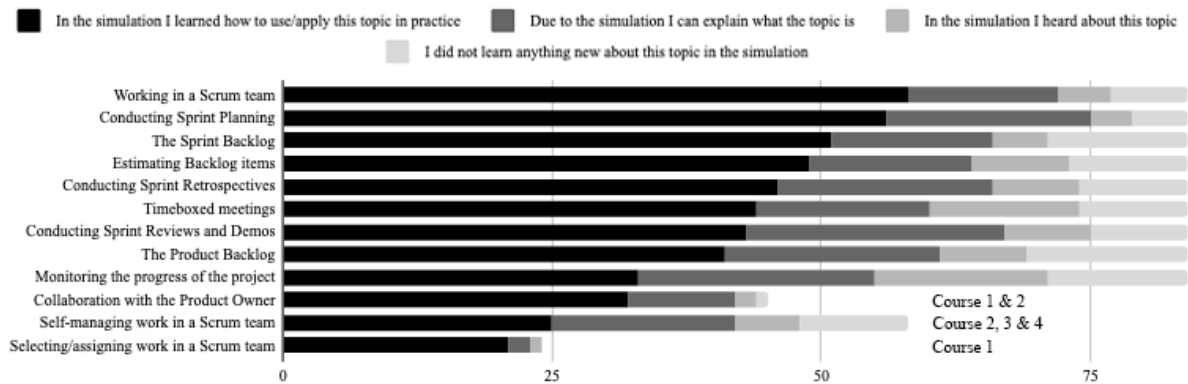


Figure 6: Scrum and agile topics

game before. However, in a development-environment I think that it will be a similar situation, as we might be working with something that we don't know a lot about.” — Student developer C, Course 2

“The choice of using Don't Starve Together as a tool for learning Scrum was, in my opinion, a genius idea. The game, being in the survival genre, is already structured in a way that encourages teamwork, as you have a better chance of survival by working together and sharing items. Beyond that, the crafting system provides easily discernible backlog items for a team to work on. It ended up working very well, and many findings from the simulation could be converted to a general software development project.” — Student developer D, Course 2

Out of the 165 students that participated in the simulation with their industrial POs, 56% (92/165) reported that communication and collaboration with the PO was a main point learned, particularly in settings where the PO does not have a technical background or a deep understanding of requirements and dependencies, as this was the scenario presented in the DST Scrum simulation:

“Our PO has no technical background. This also showed, when we played DST. I believe DST simulated quite well how we need to approach our PO to get him to understand what we mean. For instance, he had no knowledge on the game and how to survive. We needed to guide him in the beginning, such that everybody had fun and he still felt like he had a say. Our PO really wanted a pet rabbit. But this was not optimal to have as a backlog item in the first sprint, so we suggested that we move it to a sprint, where we had the foundations for both surviving and using the time needed to catch a rabbit.” — Student developer E, Course 2

Regarding the Scrum events, which were reported as a main learning by 41% of the students (81/196), Sprint Planning in particular was mentioned by a number of students who learned why this event is necessary and how the Sprint Goal can help ensure that the team focuses on the most valuable work:

“The main point I learned in the simulation was that one should formulate a clear goal at the start of sprint planning. At work, the flow has been to take tickets into the sprint and lastly define the goal. The order in the simulation seemed logical as the goal should logically guide the selection of backlog items. Also, the goal should be independent of the backlog items, i.e. a goal should not be to complete tickets A, B and C but rather to achieve a more general goal. Concretely, a good goal in the simulation could have been to make the camp livable, not to create a campfire.”

— Student developer F, Course 3

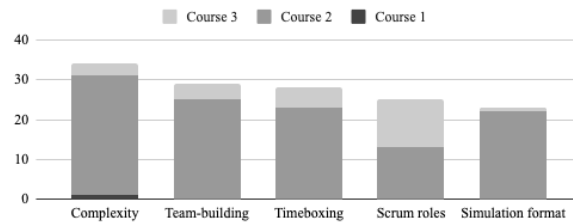


Figure 7: Student experiences and perceptions

As can be seen in Figure 5, the above presented five main topics learned were followed by work organization (79/196), teamwork (66/196), and prioritization (52/196).

5.1.2 *Survey Results.* In this section we have compiled the 84 responses to the DST Scrum simulation survey question group about learning Scrum and agile topics. The results indicate that the majority of the students learned how to use or apply the topics in practice, could explain what the topics were due to the simulation, or had heard about the topics in the simulation (Figure 6). The number of responses indicating that the students did not learn anything new about the topics ranged from zero to 18% (0-15/84).

5.2 Experiences and perceptions

In this section, we answer RQ2: “What were the experiences and perceptions of participants on the DST Scrum simulation?”

5.2.1 *Learning Diary Results.* During the data analysis some perceptions on the simulation format and the application of agile and Scrum in practice were identified in the student learning diaries (Figure 7).

Regarding the digital playground, 17% (33/196) of students reported that they thought DST was too complex. While this could indicate that DST is too complex as a play space, particularly for participants with minimal video game literacy, the level of complexity was also discussed by numerous students as one of the positive aspects of the simulation:

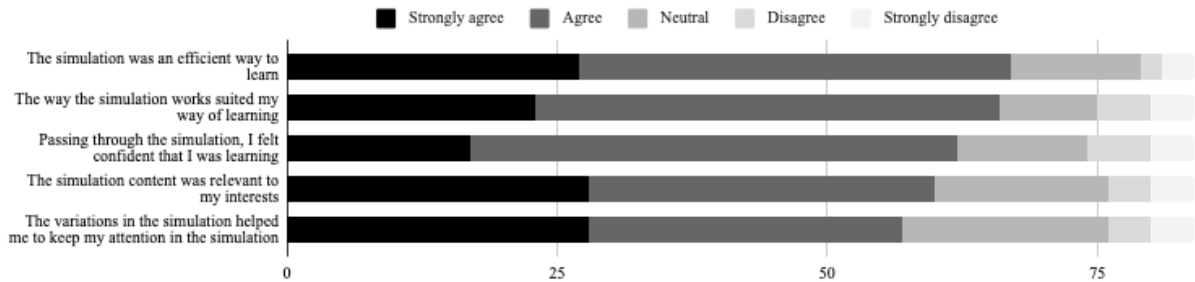


Figure 8: Motivation and learning

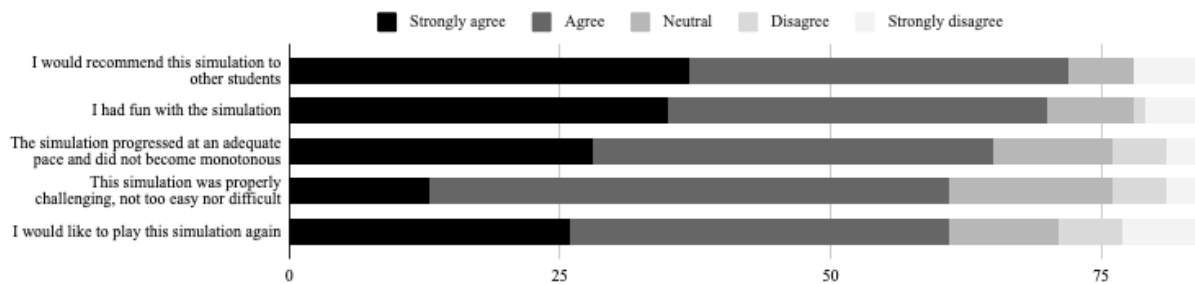


Figure 9: User experience

“As per the agile manifesto, we need to be able to respond effectively to change. In this case that means recognizing that the Sprint Backlog is too ambitious, then, in collaboration with the PO, agree and determine new priorities for the [Sprint Backlog] and finally adapt and implement necessary structural changes to meet the Sprint Goal (which may or may not have changed).”
— Student developer G, Course 2

“I feel like DST is the right game to use. Few people have tried it. It can be punishing like accidentally burning all your items. This could happen in projects as well, PO finding out that the product should go in another direction wiping all progress.”
— Student developer H, Course 2

“The game [DST] has elements of the stressful and the unknown much like a new programming project and relies on the team members working together. The game even teaches how to make use of members’ individual skill sets, which is represented with the different characters.”
— Student developer I, Course 2

This indicates that the students who thought DST was too complex had not fully grasped the complexity of software development [43] or the value of responding to change [3].

The simulation was discussed by 15% (29/196) students as a good team-building experience, both within the student group, and with the industry PO:

“As a team-building experience the simulation was excellent. We got to know our product owner and each other in a practical setting on a much deeper level, and got a taste of how effective the application of Scrum can be in a stressful setting, working with initially unknown tasks.”
— Student developer J, Course 2

Regarding the timeboxes, 14% (28/196) reported that they felt the time allotted for the Scrum events, and Sprint Planning in particular, was too short. While this could indicate that the timeboxes should

be made longer, it could also indicate a lack of understanding of Parkinson’s law, or the adage that ‘work expands to fill the time available’ [36], especially when considering that this was not an issue for the majority of participants:

“Time was a stressing factor in our simulations but it was also a lovely insight into how much more efficient you can get when using estimations and delegating work out and making sure you communicate efficiently so your team members understand you the first time.”
— Student developer K, Course 2

Regarding the Scrum roles, 13% (25/196) revealed a lack of understanding of the Scrum roles, e.g. several students were under the impression that the PO will always be a person with prior knowledge of Scrum and a thorough understanding of the technical requirements, dependencies, and domain. Finally, 12% (23/196) of students had not properly understood the possibilities and restrictions of the simulation format, e.g. several teams had only used one voice channel in Discord for all their communication, which proved to be inefficient.

5.2.2 Survey Results. The student responses to the survey question groups regarding motivation and learning are compiled in Figure 8. The responses were primarily positive, with 80% (67/84) agreeing or strongly agreeing that the simulation was an efficient way to learn. 77% (65/84) responded that the simulation suited their way of learning, while 74% (62/84) felt confident that they were learning as they passed through the simulation. The responses regarding the user experience were also primarily positive (Figure 9), with 86% (72/84) agreeing or strongly agreeing that they would recommend

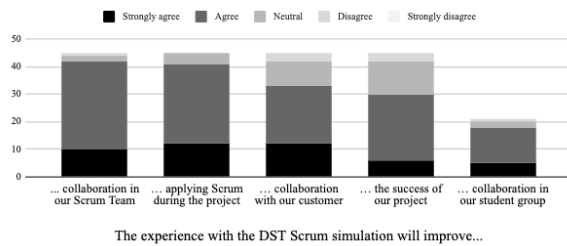


Figure 10: Impact on project

the simulation to other students and 83% (70/84) reporting that they had fun with the simulation.

The responses to the question group about the impact on the project have been compiled in Figure 10. Here the students from the first and second course were asked if the experience with the simulation would improve collaboration in their Scrum team, collaboration with their customer, the application of Scrum during the projects, and the success of their projects. The students from the second course were also asked if the experience would improve collaboration in their student group. The responses were overwhelmingly positive, with zero students strongly disagreeing and only 2-7% (1-3/45) disagreeing with the statements.

5.2.3 Retrospective. The feedback provided by the POs and agile coaches through the retrospective, showed that the simulations worked very well as a team-building activity and for practising Scrum. They reported that there was a good atmosphere, the teams had fun, and they could see how the cooperation and self-organisation of the students improved as they progressed through the simulation.

5.3 Potential Improvements

In this section, we answer RQ3: “How could the DST Scrum simulation be further improved to enhance learning opportunities?”

Our analysis showed that the simulation learning goals about agile values and principles, time management, Scrum roles and artifacts, and user stories, received less attention than expected in the learning diaries. Next, we discuss how these points could be improved, in connection with the previously mentioned experiences and perceptions.

5.3.1 Agile values and principles. Regarding agile, this learning goal could be improved in the future by emphasizing the importance of responding to change [3] in the simulation handout and introduction. As discussed in Section 1, the world is gaining complexity and students need to learn how to respond to our changing environment. This complexity and uncertainty is fully represented in DST, a point which should be communicated to participants prior to the simulation. On a similar note, the concepts of sustainability and sustainable development [4] should be promoted during the simulation. Here it may be beneficial to explain how environmental realism [10] is represented in DST, as this could both promote more sustainable development in the simulations and encourage more awareness of real-world environmental challenges.

5.3.2 Time management. It may be also be beneficial to introduce the students to Parkinson’s law [36], in connection with timeboxing and time management, prior to the simulation. This ties closely to the well established software development rule that 80% of the value in any piece of software is in 20% of the features and prioritizing by value forces teams to produce that 20% first [43]. So, while it might be beneficial to lengthen the timeboxes in the simulation to allow for more discussion during Scrum events, this could potentially result in the work expanding to fill the timeboxes and a lack of focus on the most valuable priorities in the Product Backlog. Students could also receive more explicit information about the cost of multitasking and context switching [43] and how this is represented in DST. E.g., instead of randomly collecting all the resources participants see when exploring the map, this task should ideally be taken on by specific individuals who should locate the needed resources in the biomes where they are most common. This would also help limit the amount of unnecessary resources collected, thus limiting waste [43].

5.3.3 Scrum roles and artifacts. Regarding the Scrum roles and the PO role specifically, the simulation currently simulates a scenario where the PO does not have knowledge of the domain or technical requirements and dependencies in the Product Backlog. This could potentially be improved by providing a brief training for the POs prior to the simulation where they are introduced to the basic DST game mechanics and the Product Backlog, and informed about how they could play their role to enhance learning opportunities for the students. Similar training could also be provided for the agile coaches who participate with the Scrum teams. In addition, all participants could be provided with access to the agile board, as suggested by several students. This could encourage more ownership of, and learning about the Scrum artifacts.

5.3.4 User Stories. Finally, the user story learning goal could be improved by encouraging the students to create more cards in the Product Backlog with their own user stories.

6 DISCUSSION AND CONCLUSIONS

In this paper, we presented a digital playful simulation for teaching agile, and Scrum specifically, that can be used in distributed online settings: the DST Scrum simulation. None of the previously reported games for teaching agile and Scrum (Table 1) were designed for a distributed, online teaching environment. Thus, the presented DST Scrum simulation is a unique tool for online teaching.

The DST Scrum simulation was evaluated as a tool for teaching, by collecting data from 25 Scrum teams comprised of BSc and MSc students, industry POs, and agile coaches, totalling 244 participants. The data from the 220 students was collected in the form of 84 completed surveys and 196 learning diaries. The results of this work show that the DST Scrum simulation is a valuable tool for teaching university students the Scrum framework in practice and communication in a distributed setting. The participants’ reaction to the simulation was highly positive and based on the survey responses, and the reports in their learning diaries, the students learned a lot about agile topics and the Scrum framework. The main learning outcomes included estimation, communication and

collaboration with an industry PO, Scrum events, work organisation, teamwork and prioritisation.

In addition, the inclusion of the industry PO in the first two courses provided an opportunity for team-building within the Scrum teams and many of the students reported that the experience would help improve their collaboration and the success of the projects they would do later on during the same course. When comparing the simulation learning goals to what the students actually learned, we can conclude that the students learned more than expected about work organization. Specifically, the students learned the importance of coordination and task delegation when working in a Scrum team. This became apparent through the simulations, especially when the work had not been properly delegated or communicated, which resulted in multiple students completing the same tasks.

This work has applied a CTP [1] approach, by identifying dominant discourses and normative trends in the field of SE education and testing out alternatives [25], through digital play design [40]. The dominant discourses and normative trends identified in this work are that communication is most efficient face-to-face [3], that apart from traditional teaching methods, games and gamification are most commonly used for teaching agile and Scrum [6, 7, 13, 17, 20, 21, 28, 33, 34, 45, 47], that playgrounds and play spaces are typically created as safe, simple, empowering spaces for players [40], and that teachers in SE education often act out the role of industry partners to teach students about collaboration in real-world settings [11, 34]. The results of the DST Scrum simulation show that digital communication can be efficient and effective when learned and practiced, that playful simulations can provide valuable learning opportunities for students, that complex, environmentally realistic, digital play spaces with limited supervision can be used as effective playgrounds for generating learning, and that the students benefited and learned from the involvement of the industry POs in the simulations.

One limitation of this work is that a pre-simulation data collection method was not used to assess the knowledge level of the students prior to the simulations, to be able to compare the actual learning from the simulation. As the survey participation was voluntary, the survey responses from the second course were limited, compared to the number of student participants. We aimed to mitigate these limitations through the qualitative analysis of the student learning diaries, as they were mandatory on the three first courses. However, given that the learning diary is a subjective self-assessment instrument, the main points learned could be exaggerated. Research in education finds that students tend to be overconfident in newly learned skills and overestimate the probability that their answers to general knowledge questions are correct [15]. The learning diary template provided for the students contained two examples of subheadings, which may have influenced the content. However, most of the students explained in detail what they learned and how they learned those points, by providing explicit examples of scenarios from the simulation, which indicated that they had truly understood the topics and achieved those learning outcomes.

Another limitation of this work is that the simulations in the second course were carried out by the TAs, who had varying experience in agile and Scrum, and limited experience with DST, which

affected both the playable experience and the learning opportunities for the students. The feedback collected from the industrial POs and agile coaches who participated in the simulations was also limited, so a more systematic method for collecting this data would prove beneficial in future work. This could potentially include a survey where the POs and agile coaches are requested to assess the learning outcomes of the students from the simulations, so the results could be compared with the students self-assessment. In spite of these limitations, the results of this work show how the DST Scrum simulation can provide valuable learning opportunities for students and an enjoyable digital playable experience for all participants.

REFERENCES

- [1] Philip E Agre. 1997. Lessons learned in trying to reform AI. *Social science, technical systems, and cooperative work: Beyond the Great Divide* 131 (1997).
- [2] Atlassian. 2011. *Trello [Software]*. <https://trello.com/>
- [3] Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, et al. 2001. Manifesto for agile software development. (2001).
- [4] Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, et al. 2001. Principles behind the agile manifesto. (2001).
- [5] Douglas Augusto Barcelos Bica and Carlos Alexandre Gouvea da Silva. 2020. Learning process of agile scrum methodology with lego blocks in interactive academic games: Viewpoint of students. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje* 15, 2 (2020), 95–104.
- [6] Bill Wake. 2004. Scrum from Hell. <https://xp123.com/articles/scrum-from-hell/>.
- [7] Boris Gloger. 2008. The Scrum Ball Point Game. <https://scrum4you.wordpress.com/2008/03/15/the-scrum-ball-point-game/>.
- [8] Simon Bourdeau, Alejandro Romero-Torres, and Marie-Claude Petit. 2021. Learning Scrum: A LEGO®-Scrum Simulation. In *Agile Scrum Implementation and Its Long-Term Impact on Organizations*. IGI Global, 169–189.
- [9] Daniel Cermak-Sassenrath and Charles Walker. 2012. S (t) imulating interdisciplinarity. *Simulations, Games and Role Play in University Education* (2012), 139–149.
- [10] Alenda Y Chang. 2019. *Playing nature: Ecology in video games*. Vol. 58. U of Minnesota Press.
- [11] Orges Cico, Letizia Jaccheri, Anh Nguyen-Duc, and He Zhang. 2021. Exploring the intersection between software industry and Software Engineering education—A systematic mapping of Software Engineering Trends. *Journal of Systems and Software* 172 (2021), 110736.
- [12] Rafael da Camara, Marcelo Marinho, Suzana Sampaio, and Saulo Cadete. 2020. How do Agile Software Startups deal with uncertainties by Covid-19 pandemic? *arXiv preprint arXiv:2006.13715* (2020).
- [13] Adler Diniz De Souza, Rodrigo Duarte Seabra, Juliano Marinho Ribeiro, and Lucas E da S Rodrigues. 2017. SCRUMI: a board serious virtual game for teaching the SCRUM framework. In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*. IEEE, 319–321.
- [14] Discord Inc. 2014. *Discord [Software]*. <https://discord.com/>
- [15] David Dunning, Chip Heath, and Jerry M Suls. 2004. Flawed self-assessment: Implications for health, education, and the workplace. *Psychological science in the public interest* 5, 3 (2004), 69–106.
- [16] UTS Software Engineering. 2017. Lego Scrum Game 2017. <https://youtu.be/Wy5F1XyNt74>
- [17] João M Fernandes and Sónia M Sousa. 2010. Playscrum—a card game to learn the scrum agile method. In *2010 Second International Conference on Games and Virtual Worlds for Serious Applications*. IEEE, 52–59.
- [18] Denae Ford, Margaret-Anne Storey, Thomas Zimmermann, Christian Bird, Sonia Jaffe, Chandra Maddila, Jenna L Butler, Brian Houck, and Nachiappan Nagappan. 2021. A tale of two cities: Software developers working from home during the covid-19 pandemic. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 31, 2 (2021), 1–37.
- [19] GDC. 2016. Don't Starve: Creating Community Around an Antisocial Game. <https://www.youtube.com/watch?v=0P3kFXWK07o>
- [20] Aikaterini Gkritsi. 2011. *Scrum Game: an agile software management game*. Ph. D. Dissertation. University of Southampton.
- [21] Sonja Hof, Martin Kropp, and Marla Landolt. 2017. Use of gamification to teach agile values and collaboration: A multi-week scrum simulation project in an undergraduate software engineering course. In *Proceedings of the 2017 ACM conference on innovation and technology in computer science education*. 323–328.
- [22] HpiEpicGroup. 2011. LEGO Scrum Exercise. <https://youtu.be/H2NXIDoutY>

- [23] Jamie Cheng and Kevin Forbes. 2012. Intrinsic vs. Extrinsic rewards. <https://support.klei.com/hc/en-us/articles/360035456011-Intrinsic-vs-Extrinsic-rewards>.
- [24] Jussi Kasurinen and Antti Knutas. 2018. Publication trends in gamification: A systematic mapping study. *Computer Science Review* 27 (2018), 33–44.
- [25] Vera Khovanskaya, Maria Bezaitis, and Phoebe Sengers. 2016. The case of the strangerationist: Re-interpreting critical technical practice. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. 134–145.
- [26] Klei Entertainment. 2013. *Don't Starve Together [Video Game]*. https://store.steampowered.com/app/322330/Dont_Starve_Together/
- [27] Alexey Krivitsky. 2017. Lego4Scrum 3.0: A complete guide to# lego4scrum-a great way to teach the Scrum framework and Agile thinking.
- [28] Wee Leong Lee. 2016. SCRUM-X: An interactive and experiential learning platform for teaching scrum. (2016).
- [29] Pernille Lous, Marco Kuhmann, and Paolo Tell. 2017. Is scrum fit for global software engineering?. In *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)*. IEEE, 1–10.
- [30] Matthew B Miles and A Michael Huberman. 1994. *Qualitative data analysis: An expanded sourcebook*. sage.
- [31] Courtney Miller, Paige Rodeghero, Margaret-Anne Storey, Denae Ford, and Thomas Zimmermann. 2021. "How Was Your Weekend?" Software Development Teams Working From Home During COVID-19. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 624–636.
- [32] Nathan Hook. 2018. Men identify with fictional characters of both genders. <https://www.bps.org.uk/news-and-policy/men-identify-fictional-characters-both-genders>
- [33] Emily Oh Navarro and André van der Hoek. 2004. SIMSE: An Interactive Simulation Game for Software Engineering Education.. In *CATE*, Vol. 1. Citeseer, 12–17.
- [34] Maria Paasivaara, Ville Heikkilä, Casper Lassenius, and Towo Toivola. 2014. Teaching students scrum using LEGO blocks. In *Companion Proceedings of the 36th International Conference on Software Engineering*. 382–391.
- [35] Maria Paasivaara and Casper Lassenius. 2006. Could global software development benefit from agile methods?. In *2006 IEEE International Conference on Global Software Engineering (ICGSE'06)*. IEEE, 109–113.
- [36] Cyril Northcote Parkinson and Robert C Osborn. 1957. *Parkinson's law, and other studies in administration*. Vol. 24. Houghton Mifflin Boston.
- [37] Swanky Psammead. 2019. Bound to the Board: Interpreting Theme in Klei's "Don't Starve". <https://youtu.be/3ByxB-U5yD8>
- [38] Matt Ratto. 2011. Critical making: Conceptual and material studies in technology and social life. *The information society* 27, 4 (2011), 252–260.
- [39] Ken Schwaber and Jeff Sutherland. 2020. The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. (2011). Available: scrum.org (2020).
- [40] Miguel Sicart. 2014. *Play matters*. mit Press.
- [41] State Of Agile. 2021. 15th State of Agile Report. <https://stateofagile.com/#ufh-i-661275008-15th-state-of-agile-report/7027494>.
- [42] Jan-Philipp Steghöfer, Håkan Burden, Hiva Alahyari, and Dominik Haneberg. 2017. No silver brick: Opportunities and limitations of teaching Scrum with Lego workshops. *Journal of Systems and Software* 131 (2017), 230–247.
- [43] Jeff Sutherland and JJ Sutherland. 2014. *Scrum: the art of doing twice the work in half the time*. Currency.
- [44] Christiane Gresse von Wangenheim, Rafael Savi, and Adriano Ferreti Borgatto. 2012. DELIVER!—An educational game for teaching Earned Value Management in computing courses. *Information and software Technology* 54, 3 (2012), 286–298.
- [45] Christiane Gresse Von Wangenheim, Rafael Savi, and Adriano Ferreti Borgatto. 2013. SCRUMIA—An educational game for teaching SCRUM in computing courses. *Journal of Systems and Software* 86, 10 (2013), 2675–2687.
- [46] Fandom Wiki. 2021. Don't Starve Together. https://dontstarve.fandom.com/wiki/Don%27t_Starve_Together.
- [47] William Wake and Mike Cohn. 2007. The Scrum Game. http://alecoledeLavie.com/accueil/DossierScrum/6_ScrumGame/Scrum%20Game/Scrum%20Game%20Rules.pdf.