
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Heikkilä, Ville; Paasivaara, Maria; Lassenius, Casper

Teaching university students Kanban with a collaborative board game

Published in:

Proceedings of the 38th International Conference on Software Engineering Companion

DOI:

[10.1145/2889160.2889201](https://doi.org/10.1145/2889160.2889201)

Published: 24/05/2016

Document Version

Publisher's PDF, also known as Version of record

Published under the following license:

CC BY-NC-ND

Please cite the original version:

Heikkilä, V., Paasivaara, M., & Lassenius, C. (2016). Teaching university students Kanban with a collaborative board game. In *Proceedings of the 38th International Conference on Software Engineering Companion* (pp. 471-480). IEEE. <https://doi.org/10.1145/2889160.2889201>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Teaching University Students Kanban with a Collaborative Board Game

Ville T. Heikkilä
Aalto University, Department
of Computer Science
PO Box 15400
FI-00076, AALTO, Finland
ville.t.heikkila@aalto.fi

Maria Paasivaara
Aalto University, Department
of Computer Science
PO Box 15400
FI-00076, AALTO, Finland
maria.paasivaara@aalto.fi

Casper Lassenius
Aalto University, Department
of Computer Science
PO Box 15400
FI-00076, AALTO, Finland
casper.lassenius@aalto.fi

ABSTRACT

Kanban is a workflow management method especially suitable for managing continuous software engineering work. We attempted to teach Kanban and lean thinking in a software project management course in Aalto University with a collaborative Kanban board game. Our goal was to measure if the learning goals of the class were reached and to study the student's perceptions of the game. Data was collected from two subsequent classes in 2014 and 2015. Quantitative data was collected with questionnaires and analysed descriptively and statistically. Qualitative data was collected from 57 learning diaries. The students perceived they had learned substantially from the game. They also evaluated the game very positively. However, the qualitative results and the measured learning indicated that the learning goals were only partially reached. The enjoyable game experience did not fully translate into effective learning.

CCS Concepts

•**Social and professional topics** → **Software engineering education**; Adult education; •**Software and its engineering** → *Software development methods*;

Keywords

Kanban, serious games, games-based learning, software engineering education

1. INTRODUCTION

The Kanban workflow management system has been employed in different industries since its popularisation as a part of the Toyota Production System [14], which is considered the progenitor of lean thinking. Agile methods, such as Scrum [12] and Extreme Programming [1], and lean thinking [18] have become increasingly popular in the software industry since the early 21st century [3]. Agile and lean

software development are managed quite differently from traditional projects and these contemporary methods need to be taught to software engineering students. Subsequently, in 2014, we decided that the lean principles and an application of the Kanban workflow management system [10] need to be taught to the students of a master's level software project management course in Aalto University, Finland.

One of the authors played the GetKanban¹ physical board game in a conference and found it enjoyable and effective for learning. Subsequently, we decided to employ a slightly modified version of the GetKanban v4.0 game to complement our traditional lecture and reading material based teaching. In order to evaluate the effectiveness of the game to teach Kanban, we studied the learning effects of the game both qualitatively and quantitatively during two course instances in 2014 and 2015. We also asked the students to subjectively evaluate the game and their learning from it.

This paper presents the modified GetKanban game, its evaluation and the results of students' learning. We also discuss the problems identified in the game, their causes and possible improvements to the game. The paper is structured as follows: Section 2 presents a brief introduction to Kanban and educational games research. Section 3 describes the GetKanban game and the changes we made to the original v4.0 version of the game. Section 4 presents the learning goals we aimed to reach with the game. Section 5 describes the research goals, data collection methods and the data analysis methods. The results and analysis are presented in Section 6 and discussed in Section 7. Finally, Section 8 concludes the paper and presents aims for future work.

2. BACKGROUND AND RELATED WORK

Kanban [10] is a workflow management system best known as a part of the Toyota Production System [14]. In its barest form, Kanban is based on physical Kanban cards that control the amount of inter-process inventory, also known as work-in-progress (WIP). There are many different implementations of Kanban for software development, but a Kanban system should, at minimum, visualize the workflow, limit the work-in-progress and measure the average lead time [10]. In software engineering, Kanban is considered especially suitable for managing the flow of unpredictable work such as maintenance and support tasks [18]. Typically, in software engineering, Kanban is implemented by a Kanban board which displays task cards, WIP-limits and different stages of the development process. Hereafter, when we use the

¹<http://getkanban.com/>

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored.

ICSE '16 Companion, May 14 - 22, 2016, Austin, TX, USA

© 2016 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-4205-6/16/05.

DOI: <http://dx.doi.org/10.1145/2889160.2889201>



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs International 4.0 License.

name Kanban, we refer to the board based Kanban in software engineering instead of the physical production system Kanban.

Studies on Games Based Learning, or educational games, have found positive learning effects in a variety of fields [8]. Previous research has proposed and evaluated games for software engineering education [8]. However, the quality of the evaluation, in most cases, is quite low and is typically based on the students' subjective evaluation [8]. The results of the rare experimental evaluation studies suggest that the students often perceive games enjoyable and highly educational, but the results also suggest that the games are not actually more effective than the traditional methods of learning [8].

The research on educational game design suggest that educational games should be fun, engaging, challenging and interactive [8]. The complexity of educational games needs to be carefully balanced in order to teach the subject matter without overwhelming the player with rules and mechanics [8]. There is some indication that cooperative game design may be more effective than competitive design in teaching complex subjects, especially when less and more experienced students play together and collaborate [8].

Although the knowledge of agile and lean software engineering methods is increasingly important for software engineering practitioners [15], only a few games have been proposed to teach these subjects [8]. No previous scientific articles on educational games for Kanban were found at the time of writing, although many games can be found on the internet, for example at <http://tastycupcakes.org/>.

3. THE MODIFIED GETKANBAN GAME

GetKanban v4.0 is a collaborative, physical board game that is aimed to teach the basics of Kanban. According to the game creator, the optimal number of players is from three to six. The overall goal of the game is to produce financial value, which is mainly done by gaining new subscribers, which in turn is achieved by producing new features. In order to produce features most efficiently, the players must decide how to assign workers to the features and control the WIP. When multiple teams are playing the game in parallel, the teams compete on the best outcome of the game.

Following playtesting by the course staff, the rules of GetKanban v4.0 were somewhat modified in order to make the game simpler, smoother and faster. System improvement tickets (work items) were removed. Several special events that introduce additional rules and mechanisms were removed or simplified. Initial restrictions on the gameplay were added in order to emphasize learning when the restrictions were lifted later on in the game. This section describes the modified version of the game. In the rest of this paper, the name GetKanban refers to the modified version, and all results and discussion are only related to the modified version. For the original rules of GetKanban v4.0, see <http://getkanban.com/>.

There are two modes of play, which are the standard mode and the advanced mode. The standard mode is meant for students with no Kanban experience and who need to learn the basics. The advanced mode is meant for students with pre-existing Kanban knowledge. It introduces additional mechanisms that are supposed to teach more advanced lean thinking. The differences are discussed in more detail below.

Four members of the team have a special role. One is the project manager who has the final say in decision making. One updates a cumulative flow diagram which shows the

number of tickets (work items) in each phase over time [9]. One keeps the tally of the slack (i.e. unspent work effort). Finally, one updates a financial tracking chart.

Figure 1 illustrates the Kanban board used in GetKanban. The advanced gameplay mode uses three types of tickets that represent work items. The first type is normal tickets (1) that represent new features. The second type is fixed delivery date tickets (2) that create a benefit or prevent a loss if they are completed by the delivery date. The third type is expedite-tickets that are created by event cards and have special rules. All tickets require a pre-defined amount of work on each of three main phases. The backlog contains new normal tickets and new fixed delivery date tickets. In the standard gameplay mode, only normal tickets are used.

The board has six columns (3): Ready, Analysis, Development, Test, Ready to Deploy and Deployed. The columns represent the different phases in a software development process. The Ready, Analysis, Development and Test columns have WIP limits (4). The total number of tickets in these columns cannot exceed the WIP limit. The Analysis and Development columns have the internal columns In Progress and Done (5). When work on a ticket on the internal In Progress column of the Analysis or Development column is completed, it is moved to the corresponding internal Done column. The expedite tickets are always on the Expedite lane (6) which has initial WIP of one. Tickets on the lane ignore the column WIP limits and are moved right as soon as the work on them is completed.

The available human resources are represented by dice. Each dice has a work type (Analyst, Developer, Tester) which corresponds with a main phase. The dice are the most efficient when assigned to work in the phase of their work type. When assigned to other phase, the dice are only one third efficient in comparison to their main phase.

The game is played in rounds which represent a working day. To speed the game up, the game begins on Day 8, which means that several tickets are on the board and some work on those tickets has already been completed. To clarify the game rules, the first day, Day 9, is played following a script from the game instructor. The game ends after Day 24. At the beginning of each day, the team decides how to assign the dice to the tickets. At the beginning of the game, the dice can be only assigned to their work type phase. Any number of dice can be assigned to a single ticket. After all dice have been assigned, the dice are rolled. The points given by the dice are reduced from the work left on the corresponding phase on the ticket. Any surplus points (i.e. slack) are recorded. When all work on a ticket in the Analysis or Development phase is completed, it is moved to the Done column of the phase. Starting from the Ready to deploy column, tickets are moved to the right if there are less tickets than the WIP limit in the column to the right (7).

The billing cycle of the game is three days. The first billing is done at the end of Day 9, the second at the end of Day 12 and so on. The following actions are only taken during the billing days. First, normal work item tickets and, in the advanced gameplay mode, fixed delivery date tickets are added from the backlog to the Ready column until it is filled up to the WIP. The day the tickets were added to the ready column is recorded on each ticket. Second, all tickets on the Ready to deploy column are moved to the Deployed column. The day the tickets were deployed is recorded on each ticket and the lead time is calculated and recorded on

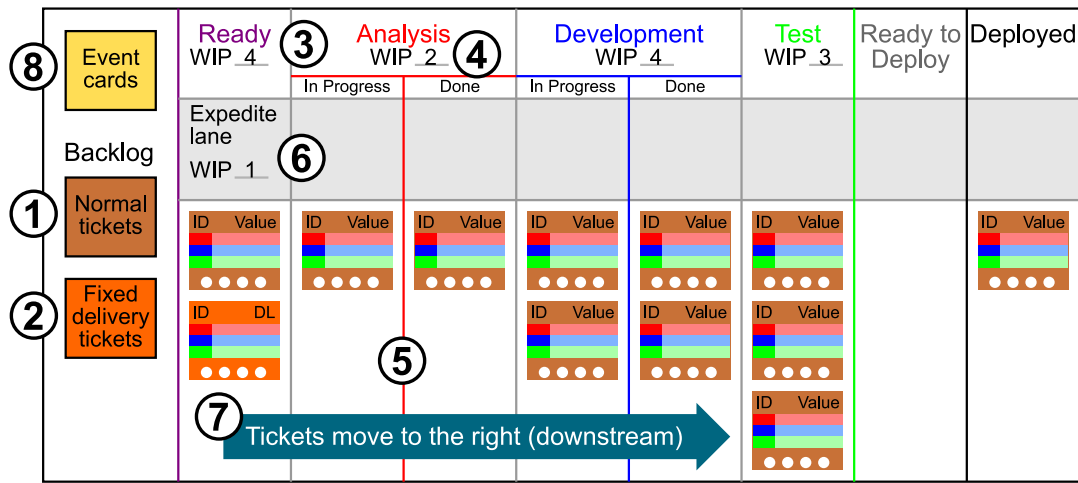


Figure 1: An illustration of the GetKanban game board

the ticket. Each deployed normal ticket adds new subscribers. The number of new subscribers is calculated with a formula based on the lead time and ticket value. The financial tracker then records the total number of subscribers and calculates the produced total value with a simple formula. The next step, done every round, is that the CFD tracker updates the cumulative flow diagram (CFD).

The final step of each round is the reading of the daily event card (8). The daily event cards contain reminders of game rules, gameplay tips and changes to the game rules. Table 1 present the event cards that change the game rules. The motive behind each rule change is discussed in the Learning goals section below.

4. LEARNING GOALS

The Kanban class included four learning methods, which were the reading material, the lectures, the GetKanban game and the post-game discussion. With these methods, we attempted to get the students to reach eight learning goals that we viewed as the core aspects of Kanban and Lean thinking. These were divided into primary (G1—G3) and secondary (G4—G8) learning goals.

The first primary learning goal (G1) was the Kanban process and the overall structure and mechanics of a typical Kanban board in software engineering. The second learning (G2) goal was to learn how to draw and interpret a cumulative flow diagram (CFD), which is a core mechanism for tracking the overall progress in Kanban. These subjects are taught by the base GetKanban game.

The third learning goal (G3) was the effect of bottlenecks on the flow, and how bottlenecks can be identified, alleviated or removed. The initial state and WIP limits of the game are designed to create a bottleneck in the Test phase. If the CFD is drawn right, the bottleneck is visible as a large space between the testing and development lines. After Day 12, all resources can be used in all phases although only one third as efficiently as in their main phase. This teaches the usefulness of the flexible, so-called T-shaped people. In addition, it demonstrates the effect of massive resource re-allocation to mitigate a bottleneck (a.k.a. swarming).

The fourth overall (G4) and the first secondary learning goal was the effect of batch processing on the flow and lead

Table 1: The rule changes given in the daily event cards.

Day	Rule change
12	The process consultant has managed to convince your managers to allow employees to work outside their specialization area. From now on, you can allocate dice to work outside the dice's specialization area. In that case, use the small number that has the color matching to the specialization area.
15 _S	The process consultant has convinced your product manager to separate the billing cycle from the planning and publishing cycle. From now on, fill the Ready queue to the WIP limit from the backlog immediately when there is space AND move tickets straight to Deployed when the testing is completed. Financial calculations still follow the 3-day billing cycle.
15 _A	A large subscriber company has requested that you tailor the system for them by the end of Day 18 and agreed to pay \$4,000 if you succeed. Ask the facilitator for Expedite ticket E1. Expedite tickets use the Expedite lane. The ticket on the expedite lane does not count towards the normal WIP. You may have maximum of one ticket on the expedite lane.
16 _A	Same as 15 _S .
18	The process consultant is now convinced you understand how WIP limits affect queues. From now on, you can change the WIP limits in the beginning of each day (before allocating dice to tickets).
20	Your engineers have been working on test automation during their slack time. You get the following benefit to all testing effort in the current and future tickets based on the total amount of slack to date (future slack does not count): Total slack to date < 35: nothing completed, no benefit. Total slack to date 35 — 74: automated unit tests, -1 effort (mark one dot as completed). Total slack to date >= 75: full test automation, -2 effort (mark two dots as completed).
21	The upper management has finally recognized the lack of resources in the testing department. Ask the game facilitator for an additional tester (green) dice.

S Only in the standard mode.
A Only in the advanced mode.

time. The three-day billing and release cycle is shown on the CFD as flat Ready and Deployed lines between the release days. After Day 15 in the Standard mode and Day 16 in the Advanced Mode, batch processing is eliminated. On CFD, this is visible as smoother Deployed and Ready lines and after a few rounds, also as the reduced average lead time.

The fifth learning goal (G5) was the effect of WIP limits on the lead time and bottlenecks. After Day 18, all WIP limits can be changed. Although swarming somewhat mitigates the bottleneck at the Test phase, reducing the WIP of incoming work is more efficient. Thus, the teams should reduce the WIP limits upstream from the Test phase. WIP changes are also reflected on the CFD as a smaller total WIP and, after a few round, as the shorter average lead time.

The sixth learning goal (G6) was different work scheduling algorithms and their applicability in different kinds of situations. The algorithms presented during the mini-lectures were FIFO, Round-Robin, Most value first, Shortest job first, Weighed shortest job first, Multi-level FIFO and Multi-level FIFO with escalation [11]. While work scheduling algorithms are not a part of the Kanban method, the game offers an opportunity to try out different methods and observe their effects. During the game, the teams make decisions on how to assign the dice to the tickets. The decisions can be made ad hoc or based on an explicit work scheduling algorithm.

Similar to work scheduling algorithms, Little’s Law (G7) [7] is not a part of the Kanban method or the GetKanban game, but it is an important concept in queue theory and lean thinking. The students were expected to learn how to calculate lead time based on Little’s Law and understand the connection between WIP, processing rate and lead time, i.e. that $lead\ time = WIP / processing\ rate^2$.

In real queues, slack helps to mitigate the effect of surges in the incoming work [11]. However, the GetKanban game does not have mechanics for this effect. Instead, the goal is to illustrate the positive aspects of slack (G8) by simulating system improvements created during the slack time. Sufficient total amount of slack by day 20 reduces the testing effort of all ongoing and incoming work.

5. METHODS

5.1 The research goal and questions

Our overall goal is to evaluate the effectiveness and efficiency of the GetKanban game and the associated lectures to teach Kanban and lean thinking. Specifically, we aim to answer to the following research questions:

RQ1 Are the learning goals (described in the previous section) reached?

RQ2 How is the game perceived by the students?

5.2 Data collection

Data was collected using questionnaires and learning diaries written by the students of the software project management course. The course was primarily aimed for master’s and doctoral level students, but bachelor students could also participate if they filled the course prerequisites, which were basic knowledge of software engineering and traditional

²This is a simplification, see Little [7] for a detailed description of Little’s Law.

Table 2: The true/false statements and associated learning goals

ID	Statement	LG*
Q2.1	WIP limits upstream from a bottleneck should be decreased	G3, G5
Q2.2	Kanban is a push-based workflow management system	G1
Q2.3	The goal of Kanban is to maximize resource utilization	G1
Q2.4	The goal of Kanban is to minimize lead time	G1
Q2.5	A cumulative flow diagram shows the WIP over time	G2
Q2.6	A cumulative flow diagram shows the average lead time	G2
Q2.7	A cumulative flow diagram shows bottlenecks	G2, G3
Q2.8	Specialization increases resource utilization	G3
Q2.9	Slack is always a form of waste	G8
Q2.10	Large batches increase throughput	G4
Q2.11	Large batches increase lead time	G4, G5, G7
Q2.12	Large batches increase WIP	G4, G7
Q2.13	Round-robin scheduling prevents starvation	G6
Q2.14	FIFO is the most efficient work scheduling algorithm	G6

*Learning goal(s)

project management. In total, 31 and 43 students participated in the course in 2014 and 2015, respectively. All students did not participate in all classes or complete the course. In 2014, there were four bachelor level students and no doctoral level students. In 2015, there were four bachelor level students and one doctoral level student. The rest of the students were on the master’s level.

The students filled out three questionnaires³ related to this research. The first questionnaire was filled by the students before the first lecture of the course. The second, pre-game questionnaire was filled by the students at the beginning of the GetKanban game class and the third, post-game questionnaire at the end of the class. The purpose was to evaluate the effects of the class to the students’ knowledge and to evaluate the game class.

The pre-course questionnaire contained questions for demographic information and for the self-evaluation of the subjects taught during the course. The knowledge evaluation scale was loosely based on the Bloom’s taxonomy of learning [4]. The pre-course self-evaluation items are shown in Table 4. The self-evaluation scale ID_A and the associated Bloom’s taxonomy levels are shown in Table 3.

The pre-game questionnaire was filled before the GetKanban game. It contained Kanban and Lean related true/false questions and also had the answer option ”I don’t know”. Table 2 lists all true/false questions and the associated learning goals. The questionnaire also contained questions for the self-evaluation of knowledge on Kanban and Lean related topics. The full list of topics is shown in Table 5. The self-evaluation used the scale ID_B shown in Table 3. The purpose of the pre- and post-game questionnaires was to evaluate the effect of the GetKanban game and the associated mini-lectures.

In 2014, GetKanban was played by five teams consisting

³The questionnaires are available in a web appendix at <https://wiki.aalto.fi/x/zJevBg>

Table 3: The self-evaluation scale

ID_A	ID_B	Scale item	Bloom's level
1	1	I have no knowledge on the topic	No knowledge
2	2	I have heard of the subject	Remembering ₁
3	—	I have some knowledge on the topic*	Remembering ₂
4	3	I can explain what it is	Understanding
5	4	I have used/applied it in practice	Applying
6	5	I have made/suggested improvements to it in practice	Analysing / Evaluating / Creating

* Only in the pre-course questionnaire

of five to six students each. Two teams played the advanced mode and three the standard mode game. The students were assigned to the teams based on the pre-course self-evaluation on Kanban and Lean related topics in a such way, that the most knowledgeable students were assigned to the advanced mode teams. In 2015, the set-up was identical to 2014 except that all teams played the advanced mode, as we aimed to teach all students both the basic and advanced topics.

Before the game class, the students were assigned to read a short introduction to Kanban [10]. In 2014, two hours were reserved to the game and in 2015 three hours. Both classes were begun with the pre-game questionnaire. Ten minutes were given to filling the questionnaire. Before the game started, the lecturer presented an introduction to the game and its rules in a fifteen-minute presentation. In 2014, the game was interrupted two times during the class to give a short lecture on topics related to the game. The first lecture was given approximately 45 minutes after the game had started. It gave the students information on Little's Law [7], slack, resource utilization, WIP, T-shaped people and swarming. The second lecture was given approximately one and a half an hour after the game had started. It contained information on work scheduling algorithms. Both presentations lasted approximately 10 minutes. In 2015, both lectures were combined and given in the class one week before the GetKanban game class. None of the teams reached Day 24 during the 2014 class. The teams reached the day 15, 18, 19, 21 and 22. During the 2015 class, one team reached Day 21 and the rest four times reached Day 24.

After the time reserved for the game had ended, there was a short, 15 minute discussion on the game. Following that, the students had 15 minutes to fill the post-game questionnaire. It contained the same questions and statements as the pre-game questionnaire and, in addition, had several statements to evaluate the GetKanban game based on the evaluation model proposed by von Wangenheim et al. [16, 17]. The evaluation model was selected because it had been employed in previous games-based learning studies and validated by previous research [16, 17].

The evaluation was done using a Likert-type scale with the following labels: Strongly disagree, Disagree, Neither agree or disagree, Agree and Strongly agree. The full list of statements evaluated by the students is shown in Figure 2. Questions Q4.1–Q4.10 evaluate the motivation of the students to play the game and questions Q5.1–Q5.14 evaluate the game user experience. Q6.1–Q6.3 evaluate the perceived learning from the game.

After the class, the students were given one week of time to write a 700 to 1400 word learning diary entry on the game and the class in general. To obtain full marks from the learning diary entry, the students were instructed to

reflect on their learning, discuss and compare the game with other learning materials, to discuss the game as a format of learning and give feedback on the game. In total, 27 learning diaries were received in 2014 and 30 in 2015.

5.3 Data analysis

The differences in the pre-existing knowledge of Kanban and Lean related topics between the years were analysed with Independent Samples Mann-Whitney U Test. The changes in the true/false statements (Q2.1–Q2.14) between the pre-game and post-game questionnaires were analysed with the Sign Test. The changes between the pre-game and post-game self-evaluation questionnaires (Q3.1–Q3.9) were analysed with Wilcoxon Signed-Rank Test. The game evaluation statements (Q4.1–Q6.3) were analysed descriptively.

The learning diary entries were imported to the qualitative analysis program Atlas.ti. The following code list was created based on the research questions: One code for each of the eight learning goals, "Bad thing in the class", "Good thing in the class" and "Game improvement suggestion". The diaries were read in their entirety and coded. In total, 320 passages of text were coded and analysed. All passages of text coded under each code were extracted and analysed code by code. The qualitative results were written based on this analysis.

6. RESULTS AND ANALYSIS

6.1 Quantitative results and analysis

The first analysis step was the identification of possible differences in the background knowledge of the students between the 2014 and 2015 classes. Table 4 shows an overview of the students' background knowledge self-evaluation results. See the scale ID_A in Table 3 for the scale. There were no statistically significant differences between the years in the items Q1.1–Q1.4. The item Q1.5 (Lead time) showed a statistically significant difference. The distribution of answers was different and the 2015 data had a step higher median answer to Q1.5. Descriptive analysis of the students' backgrounds did not explain this difference. In subsequent analyses, the two years were combined except for the lead-time related statements Q2.4, Q2.6 and Q2.10.

The next step was the analysis of the differences between the pre- and post game questionnaires in order to measure perceived learning. The answers to all knowledge self-evaluation items improved statistically significantly between the pre- and post-game questionnaires. Applying the Holm-Bonferroni correction method for multiple comparisons ($N = 9$) [5] does not alter the significance results. Table 5 shows the medians, inter-quartile ranges and p-values of the items pre- and post game. See the scale ID_B in Table 3 for the scale.

Table 4: Descriptive statistics of the students’ background knowledge ($N_{2014} = 30$ ($Q_{1.4} = 29$), $N_{2015} = 41$) (scale ID_A in Table 3)

ID	Topic	\tilde{x}_{2014} (Bloom’s)	\tilde{x}_{2015} (Bloom’s)	IQR_{2014}	IQR_{2015}	p^*
Q1.1	Kanban	3 (Remembering ₂)	3 (Remembering ₂)	3	2	0.981
Q1.2	Lean principles	3 (Remembering ₂)	3 (Remembering ₂)	2	1.5	0.674
Q1.3	Lean software development	3 (Remembering ₂)	3 (Remembering ₂)	2.25	1.5	0.439
Q1.4	Work-in-progress	3 (Remembering ₂)	3 (Remembering ₂)	2	3	0.301
Q1.5	Lead-time	2 (Remembering ₁)	3 (Remembering ₂)	2	3	0.048

*Mann-Whitney U, 2-tailed

Table 5: Results of the knowledge self-evaluation ($N = 59$) (scale ID_B in Table 3)

ID	Topic	LG*	\tilde{x}_{pre} (Bloom’s)	\tilde{x}_{post} (Bloom’s)	IQR_{pre}	IQR_{post}	p^{**}
Q3.1	The structure of a Kanban board	G1	3 (Understanding)	4 (Applying)	1	0	< 0.001
Q3.2	Cumulative flow diagram	G2	2 (Remembering)	3 (Understanding)	1	1	< 0.001
Q3.3	WIP limits	G3, G5	3 (Understanding)	4 (Applying)	1	1	< 0.001
Q3.4	Identifying bottlenecks in a development flow	G3	2 (Remembering)	3.5	1	1	< 0.001
Q3.5	T-shaped people	G3	1 (No knowledge)	3 (Understanding)	2	1	< 0.001
Q3.6	Work scheduling algorithms	G6	2 (Remembering)	3 (Understanding)	1	1.75	< 0.001
Q3.7	Little’s law	G7	1 (No knowledge)	2 (Remembering)	1	1	< 0.001
Q3.8	One piece flow	G4	1 (No knowledge)	2 (Remembering)	1	2	0.001
Q3.9	Swarming	G3	1 (No knowledge)	2.5	1	1	< 0.001

*Learning goal(s)

**Wilcoxon Signed Ranks Test, 2-tailed

In order to measure the actual learning, the changes in the true/false statements between the pre- and post game questionnaires were analysed with the Sign Test. The results show, that the answers to most of the true/false statements did not change in a statistically significant way. Table 6 shows the changes to the true/false statements between the pre- and post test and the results of the Sign Test. The answers to Q2.5, Q2.7, Q2.9 and Q2.13 show a statistically significant change on the $p < 0.05$ level. However, when corrected for multiple comparisons ($N = 14$) using the Holm-Bonferroni method [5], none of the results are statistically significant.

The final analysis was the students’ perceptions of the game. The results show that the game was, in general, highly motivating (Q4.1—Q4.10), had a good user experience (Q5.1—Q5.14) and was considered a good learning method (Q6.1—Q6.3). The game evaluation results are shown in detail in Figure 2.

6.2 Qualitative results and analysis

In general, the GetKanban game was very positively received and seen as fun, immersive and educationally effective: “... I do appreciate and enjoy this Kanban simulation game. I obtain a lot from the game, with myself immersing in a funny and harmony atmosphere cooperating with my team, competing with other teams.” Especially the team work elements and the tactile and visual nature of the game were perceived as the positive aspects of the game. A few students wrote that the game allowed experimentation with the Kanban process parameters without endangering a real project. Furthermore, some students would have liked to play the game again to test different strategies. Most of the feedback

on the visual design of the game was positive.

The biggest issue the students identified in the 2014 game was the lack of time. In the 2015 diaries, one common problem identified by the students was the size of the teams playing the game (5—6 people), which meant that it was difficult for everyone to contribute all the time. The tracking of slack and the financial calculations were considered superfluous and not contributing to learning by some students. Some students noticed that they had concentrated on maximizing the profit and not on learning Kanban. Many students complained that the use and utility of the CFD was not properly explained during the game or after it: “I felt that the work of ... the graph drawer ended up being kind of pointless because we weren’t explicitly asked to have a look at those charts at the end of the game to come to some conclusions.” Some students thought that some abstractions in the game, such as dice throwing and the financial calculations, did not sufficiently reflect the real world. A few students found the game too difficult and a few too boring towards the end. Most students did not like the game play interruptions created by the mini-lectures in 2014.

The improvement suggestions included a longer time to play the game, giving the mini-lectures before the game (in the 2014 diaries), clarifying which parts of the game reflect a real Kanban project (“... I was unsure how many of the different parts of the Kanban board are actually present in real life examples.”), time boxing the gameplay steps in order to make the game faster and allowing the teams to change the WIP limits earlier in the game.

According to the learning diaries, most students had learned the basic structure a Kanban board and the Kanban method: “The Kanban game helped me to learn Kanban from two

Table 6: The true/false statement answer changes between pre- and post-game questionnaires (R = right answer, W = wrong answer, N = don't know)

ID	Positive changes				No change				Negative changes				p^*
	$W \rightarrow R$	$N \rightarrow R$	$W \rightarrow N$	Σ	$R \rightarrow R$	$N \rightarrow N$	$W \rightarrow W$	Σ	$R \rightarrow N$	$N \rightarrow W$	$R \rightarrow W$	Σ	
Q2.1	5	14	0	19	15	3	4	22	3	8	4	15	0.607
Q2.2	5	4	0	9	31	0	10	41	0	4	4	8	1.000
Q2.3	5	1	1	7	11	2	26	39	1	4	7	12	0.359
Q2.4** ₂₀₁₄	4	2	0	6	11	2	4	17	1	1	3	5	1.000
Q2.5	6	21	0	27	13	6	1	20	1	6	4	11	0.015
Q2.6 ₂₀₁₄	1	3	0	4	1	8	3	12	1	10	1	12	0.077
Q2.6 ₂₀₁₅	1	6	1	8	7	2	6	15	1	3	3	7	1.000
Q2.7	1	21	2	24	14	8	3	25	1	5	3	9	0.015
Q2.8	3	3	3	9	9	7	17	33	0	9	7	16	0.230
Q2.9*** ₂₀₁₄	0	2	0	2	1	2	11	14	0	10	2	12	0.013
Q2.10	2	13	1	16	14	9	8	31	5	2	4	11	0.441
Q2.11 ₂₀₁₄	2	5	1	8	8	5	1	14	4	1	1	6	0.791
Q2.11 ₂₀₁₅	3	7	0	10	10	1	4	15	1	2	2	5	0.302
Q2.12	6	11	1	18	22	6	3	31	3	4	3	10	0.186
Q2.13	0	13	1	14	19	21	0	40	0	3	1	4	0.031
Q2.14	4	7	0	11	26	10	0	36	2	2	7	11	1.000

*The Sign Test

** Q2.4 was erroneous in the 2015 questionnaires and is omitted here.

*** Q2.9 was considered too ambiguous and was omitted from the 2015 questionnaires.

aspects: to better understand the structure and workflow in Kanban and a lot of new concepts.” Several students stated that they learned how to draw a CFD, but many would have liked to discuss the CFD in depth during the class. Many students had identified the bottleneck in the Test phase and discussed different methods for mitigating the bottleneck, including T-shaped people to swarm the bottleneck. The term batch processing was not mentioned in any learning diary. However, a few students discussed the cost of partially done work based on their real-life experiences. The purpose of WIP limits was discussed in a majority of the learning diaries. A few students stated that the only WIP limit relevant in the game was the limit of the incoming new tickets. Some students stated that they did not understand the purpose of the WIP limits. One team had decided to remove the WIP limits completely, which had resulted in massive queues in the Test column. Although work scheduling algorithms were discussed in almost all learning diaries, only a few teams used a specific work scheduling algorithm. Instead, ad-hoc scheduling and scheduling that minimized slack were the most commonly mentioned methods. Little’s Law was only briefly mentioned in a few learning diaries. The approach to slack was roughly divided into two categories in the learning diaries. The first was diaries that discussed what slack was in the real world and how it was not always harmful. The second was diaries that considered slack always harmful. Several students also complained that the slack in the game had little to do with the slack in the real world.

7. DISCUSSION

7.1 Were the learning goals reached? (RQ1)

Based on the results, the first learning goal, the Kanban process and board (G1), was only partially reached. The results of the self-evaluation of knowledge on the structure

of a Kanban board (Q3.1) and the qualitative results suggest that the students thought that they had learned the basics of the Kanban process and the Kanban board. However, the results from the true/false statements Q2.2—Q2.4 show a different trend. The game did not have a significant effect on the knowledge of pull-based workflow management in Kanban (Q2.2). Several students gave the same wrong answer to Q2.2 both in the pre- and post game questionnaires, which means that the game failed to correct false knowledge. Similarly, the game also failed to teach that the goal of Kanban is not to maximize resource utilisation (Q2.3). The results of the lead time question (Q2.4) suggest no significant learning. These results indicate that the game failed to teach that the goal was to find the optimal balance between the resource utilization and lead time.

The results on the second learning goal, the cumulative flow diagram (CFD) (G2), are more encouraging. The self-evaluation of cumulative flow diagram knowledge (Q3.2) increased by one step. The student’s understanding of how a CFD shows the WIP over time (Q2.5) and bottlenecks (Q2.7) seemed to have improved during the game although not in a strictly statistically significant way. The results regarding how a CFD shows the average lead time (Q2.6) were mixed, especially in the 2014 answers. The learning diaries indicate that especially the students who had the CFD tracker role learned how to draw a CFD, but most teams did not employ the CFD in their decision making. These results suggest that playing the game teaches how to draw a CFD and teaches some knowledge on the interpretation of a CFD.

The third learning goal, identifying and removing bottlenecks (G3), got mixed results. The improvements in the related self-evaluation questions Q3.3, Q3.4, Q3.5 and Q3.9 were the most notable of all self-evaluation questions. However, the changes in the related true/false statements were mixed. The answers to Q2.1, “WIP limits upstream from a

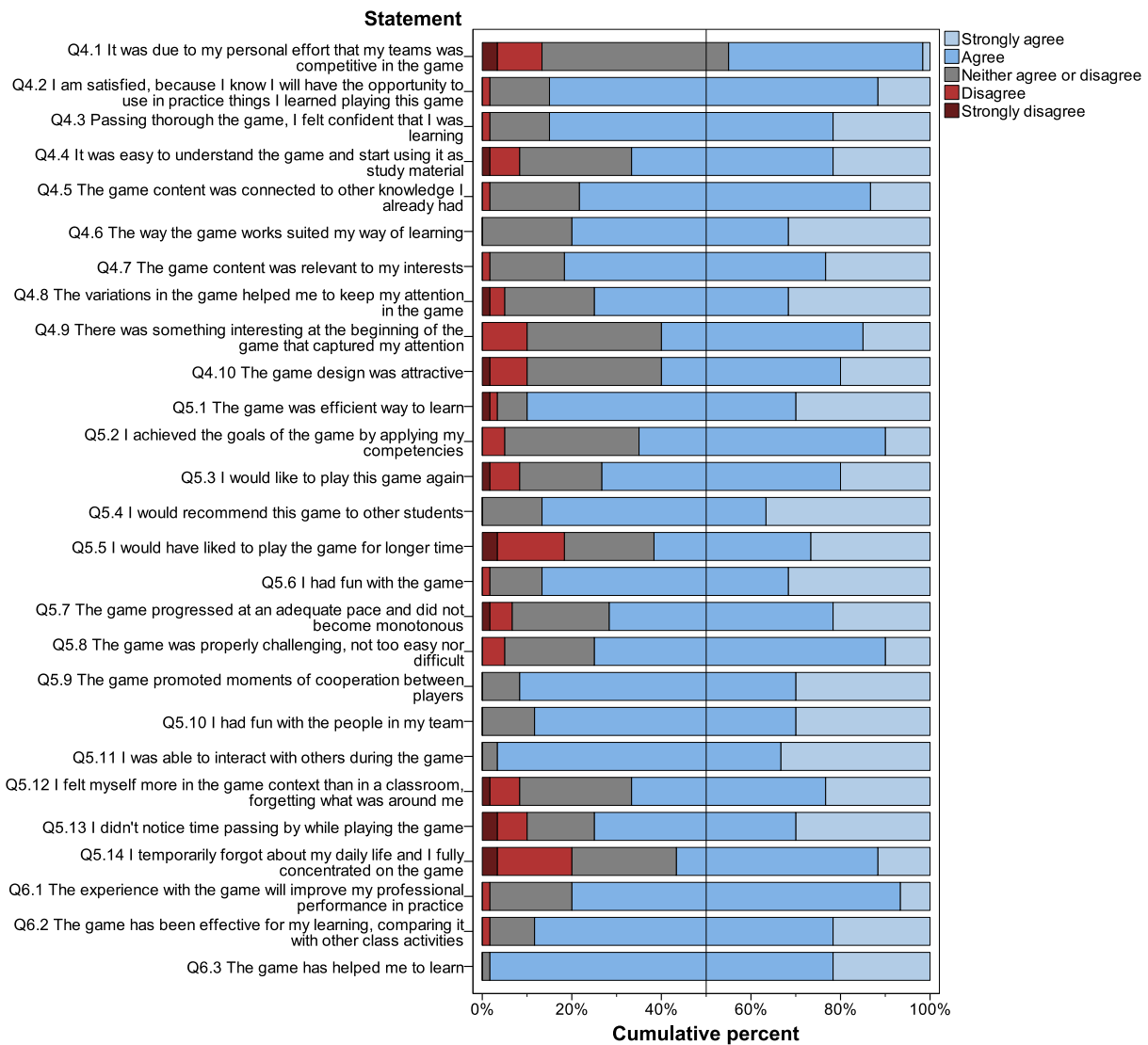


Figure 2: The results of the game evaluation statements ($N = 60$)

bottleneck should be decreased”, changed almost equally in the correct and incorrect direction. Many students seemed have learned that bottlenecks can be identified from a CFD (Q2.7) although the results is not strictly statistically significant. The results on the effect of specialisation on resource utilisation (Q2.8) did not show significant change. Most respondents (57.9 %) did not change their answer and notable portion of the respondents kept their wrong answer (29.3 %). Based on these results, methods for identifying and removing bottlenecks were not sufficiently taught by the game.

The effects of batch processing (G4) were not explicitly addressed in the game although the in- and outflows of tickets at the beginning the game were batch processes. The results indicate some learning on the topic, as the knowledge self-evaluation of one piece flow (Q3.8) increased by one step from No knowledge to Remembering. Yet, the interquartile range was quite high, indicating high variability among the students. None of the answers to the related true/false statements (Q2.10—Q2.12) changed in a significant way. The game does not seem to be well suited to teach this subject.

Tied with the structure of a Kanban board (Q3.1), the WIP limits (Q3.3) got the highest score on the knowledge self-evaluation. The median increased from 3 (Understanding) to 4 (Applying). This would suggest that the fifth learning goal, the effects of WIP limits on the lead time and bottlenecks (G5), might have been reached. Unfortunately, the results of the related true/false statements Q2.1 and Q2.12 do not show significant change. The results from the learning diaries also indicate that notable portion of the students did not understand the purpose and effects of the WIP limits. Although WIP limits are a core element of the Kanban method, the game failed to teach their significance to a considerable part of the students.

The game offers an opportunity to try out different work scheduling algorithms (G6). Although the knowledge self-evaluation (Q3.6) indicates improvement from Remembering to Understanding, the high interquartile range indicates high variability between the students. Based on the learning diaries, most teams did not use a specific work scheduling algorithm during the game. The results from the true/false

statements suggest that several students learned the basics of round-robin scheduling (Q2.13). Nevertheless, most students did not change their answer to statements Q2.13 and Q2.14 (FIFO) between the pre- and post game questionnaires. In addition, considerable part of the students answered right to those questions in both pre- and post game questionnaires, indicating pre-existing knowledge on the topic. Although the game, in theory, is suitable for simulating different work scheduling algorithms, the way it was used in the classes does not seem to be suitable for this purpose.

The cumulative flow diagram drawn during the game is well suited for illustrating Little's Law (G7) and its power to predict lead times. The median of the related knowledge self-evaluation question (Q3.7) increased one step from No knowledge (1) to Remembering (2), but the answers to the related true/false statements Q2.11 and Q2.12 did not change significantly. A discussion and analysis of a CFD after the game might be a more suitable method for teaching Little's Law instead of expecting the students to understand the connection between the game mechanics and Little's Law during the game.

The goal was to illustrate the positive aspects of slack (G8) by simulating system improvements created during the slack time. During 2014, this effect was diminished by the timing of the system improvement event, as all teams did not reach the event card on Day 20. The related true/false statement (Q2.9) was omitted from the 2015 questionnaire as too ambiguous. Due to these issues, we cannot say anything reliable about the capability of the game to teach about slack.

There is some indication that the primary learning goals G1, G2, G3 and the secondary learning goal G5 were partially reached although there was high variability between the students. The results regarding the secondary learning goals G4, G6, G7 and G8 were less positive. The game seems to have taught the basics of the Kanban method and board, but the more abstract and complex aspects related to queue theory and lean thinking were learned insufficiently or not at all. The students seem to have concentrated too much on the gameplay mechanics and winning the game instead of understanding the Kanban method and reflecting on lean thinking. The positive learning self-evaluation results are in a sharp contrast with the results from the true/false statements. Only a few of the statements showed notable improvement. In most of the statements, many of the students did not change their answer between the pre- and post-game questionnaires and the positive and negative changes were of similar size. The inconsistency between student's self-evaluation of learning and measured learning has been identified in previous SE education research [8]. Based on our results and the previous research [8], the evaluation of learning from educational games should not be based solely on student's subjective opinions, as such evaluation may be highly misleading.

7.2 How was the game perceived by the students? (RQ2)

The feedback on the GetKanban game in the learning diaries was overwhelmingly positive. The quantitative game evaluation results (Q5.4 and Q5.6) also indicate primarily positive attitudes towards the game.

The learning related statements, Q4.2, Q4.3, Q4.6, Q5.1, Q6.1, Q6.2 and Q6.3, show very positive attitudes towards

learning from the game. Although the understandability of the game (statements Q4.4 and Q5.8) got mostly positive evaluation, the statements and the learning diaries suggest that a notable portion of the students had difficulties understanding the game. This alludes to the need to further clarify the game instructions and guidance. Based on the statements Q4.5 and Q4.7, the students perceived the game as relevant to them.

The learning diaries suggested that teams of five to six people were too large and all team members did not feel they contributed to the game. The problem may be also reflected in the game immersion related statements Q5.12, Q5.13, Q5.14. While the statements got mostly positive answers, the variability was quite high and there was a notable number of negative answers.

The game rules were given mixed feedback. The statements Q4.8 and Q4.9 received notable number of neutral and negative answers. This might be explained by the variability in the background of the students. According to the learning diaries, some students with existing Kanban experience thought the game was a little too easy and boring, while a few students with little software engineering experience thought that some rules and concepts in the game were too difficult. This suggests, that dividing the teams into different gameplay modes based on their background knowledge, as was done in 2014, might be beneficial to the game user experience.

The variability in the responses to the statements Q5.3, Q5.5 and Q5.7 was quite high. This was also reflected in the learning diaries, as some students considered the game boring, while others would have liked to play the game again. This might also be a result of the varying backgrounds of the students.

According to the results of the statement Q4.10, most students considered the game design attractive, but a notable portion gave a neutral or negative answer. While this subject is highly subjective, this result and the learning diaries indicate that the game design needs some polishing.

The cooperation and interaction between the students in a team were perhaps the most positively rated aspects of the game. The statements Q5.9, Q5.10 and Q5.11 got overwhelmingly positive ratings from the students.

7.3 Threats to validity

In this section, we discuss the validity of our study based on the validity aspects described by Shadish et al. [13]. These aspects are the internal validity, construct validity, statistical conclusion validity and external validity.

Internal validity is concerned with the validity of the causal relationships identified in a study. The main goal of the statistical analysis was to identify the changes between the students' pre- and post-game knowledge. Due to the lack of a control group, we cannot rule out that the testing (i.e. the questionnaires) itself affected the students' answers [13].

Construct validity is concerned with the operationalization of the constructs employed in a study [13]. The main constructs used in this study were the *perceived learning*, the *actual learning* and the *perceptions of the game*. The perceived learning was quantitatively operationalized with the self-evaluation questions, the actual learning with the true/false questions and the perceptions of the game with the game evaluation statements. Actual learning is a complex subject and it is clear that measuring it with a few true/false

questions does not produce conclusive results. Those results should be only considered indicative of the actual learning. However, all quantitative results were triangulated [6] with the qualitative results from the learning diaries. The triangulation enabled us to hypothesize plausible explanations for the quantitative results. Although there were some differences in the student's backgrounds between 2014 and 2015, the learning results did not show statistically significant differences. Subsequently, the results of the two course years were not discussed separately.

Statistical conclusion validity is concerned with the validity of the statistical analysis methods employed in the research [13]. The statistical analysis methods employed in this research are widely accepted as robust and valid for the purposes they were used [2]. Furthermore, all p-values were corrected using the Holm-Bonferroni correction method for multiple comparisons [5]. Thus, it is not very likely that the results of the statistical analysis contain Type I errors. To mitigate Type II errors, we have taken results with low but not significant p-values into consideration when the results have been supported by the qualitative data.

Finally, external validity is concerned with the generalizability of the results [13]. Our results can be only generalized to students with a similar educational, professional and cultural background. As the data has been collected from two separate classes over two years, it is unlikely that the results reflect an edge case of our student population.

8. CONCLUSIONS AND FUTURE WORK

The results of our study and previous research indicate that educational games may not be more effective in teaching software engineering than more traditional methods. Although games do not either seem to be less effective, the efficiency of educational games in software engineering is questionable. Games introduce overhead, such as learning the rules and setting up the game and may require more time to complete than other methods. Games may also have gameplay mechanics that have the tendency to draw the players' attention away from the actual learning goals of the game.

In the future, we are going to further improve GetKanban based on the results of this research. During the software project management course in 2016, the students will play the further improved GetKanban game and we will compare the learning results to the version employed in 2014 and 2015 in order to measure if the improvements have had the desired effect.

Acknowledgements

We would like to thank Veikko Isotalo for the preliminary data analysis and Päivi Kinnunen for assisting us with the research design.

9. REFERENCES

- [1] K. Beck and C. Andres. *Extreme programming explained : embrace change*. Addison-Wesley, Boston, MA, USA, 2nd edition, 2004.
- [2] H. Coolican. *Research methods and statistics in psychology*. Hodder & Stoughton, London, UK, 3rd edition, 1999.
- [3] T. Dingsøy, S. Nerur, V. Balijepally, and N. B. Moe. A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6):1213–1221, 2012.
- [4] M. Forehand. Bloom's taxonomy. In M. Orey, editor, *Emerging Perspectives on Learning, Teaching, and Technology*. Department of Educational Psychology and Instructional Technology, University of Georgia, 2001.
- [5] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.
- [6] T. D. Jick. Mixing qualitative and quantitative methods: Triangulation in action. *Administrative Science Quarterly*, 24(4):602–611, 1979.
- [7] J. D. C. Little. OR FORUM—Little's Law as viewed on its 50th anniversary. *Operations research*, 59(3):536–549, 2011.
- [8] M. Paasivaara, V. Heikkilä, C. Lassenius, and T. Toivola. Teaching students Scrum using LEGO blocks. In *Companion Proceedings of the 36th International Conference on Software Engineering*, ICSE Companion 2014, pages 382–391, New York, NY, USA, 2014. ACM.
- [9] K. Petersen and C. Wohlin. Measuring the flow in lean software development. *Software: Practice and Experience*, 41(9):975–996, 2011.
- [10] K. Rautiainen. Kanban for software development. In V. T. Heikkilä, K. Rautiainen, and J. Vähäniitty, editors, *Towards Agile Product and Portfolio Management*, chapter 12, pages 184–192. Aalto University, Helsinki, Finland, 2011.
- [11] D. G. Reinertsen. *Principles of product development flow : second generation lean product development*. Celeritas Publishing, Redondo Beach, CA, USA, 2009.
- [12] K. Schwaber and M. Beedle. *Agile software development with Scrum*. Prentice-Hall, Upper Saddle River, NJ, USA, 2002.
- [13] W. R. Shadish, T. D. Cook, and D. T. Campbell. *Experimental and quasi-experimental designs for generalized causal inference*. Houghton Mifflin, Boston, MA, USA, 2001.
- [14] Y. Sugimori, K. Kusunoki, F. Cho, and S. Uchikawa. Toyota production system and Kanban system materialization of just-in-time and respect-for-human system. *International Journal of Production Research*, 15(6):553–564, 1977.
- [15] VersionOne, Inc. 9th annual "state of agile development" survey. <http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf>, 2015.
- [16] C. G. von Wangenheim, R. Savi, and A. F. Borgatto. Deliver! – an educational game for teaching earned value management in computing courses. *Information and Software Technology*, 54(3):286–298, 3 2012.
- [17] C. G. von Wangenheim, R. Savi, and A. F. Borgatto. SCRUMIA—An educational game for teaching SCRUM in computing courses. *Journal of Systems and Software*, 86(10):2675–2687, 10 2013.
- [18] X. Wang, K. Conboy, and O. Cawley. "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems and Software*, 85(6):1287–1299, 2012.