No Silver Brick: Opportunities and Limitations of Teaching Scrum with Lego Workshops

Jan-Philipp Steghöfer, Håkan Burden, Hiva Alahyari, and Dominik Haneberg

Department of Computer Science and Engineering, Chalmers | University of Gothenburg. jan-philipp.steghofer@gu.se and hiva@chalmers.se.
RISE Viktoria, Gothenburg, Sweden. hakan.burden@ri.se.
Institute for Software & Systems Engineering at the University of Augsburg, Germany. haneberg@isse.de.

Abstract

Education in Software Engineering has to both teach technical content such as databases and programming but also organisational skills such as team work and project management. While the former can be evaluated from a product perspective, the latter are usually embedded in a Software Engineering process and need to be assessed and adapted throughout their implementation. The in-action property of processes puts a strain on teachers since we cannot be present throughout the students' work. To address this challenge we have adopted workshops to teach Scrum by building a Lego city in short sprints to focus on the methodological content. In this way we can be present throughout the process and coach the students. We have applied the exercise in six different courses, across five different educational programmes and observed more than 450 participating students. In this paper, we report on our experiences with this approach, based on quantitative data from the students and qualitative data from both students and teachers. We give recommendations for learning opportunities and best practices and discuss the limitations of these workshops in a classroom setting. We also report on how the students transferred their methodological knowledge to software development projects in an academic setting.

Keywords: Scrum, agile software engineering, software engineering education

1. Introduction

Teaching Software Engineering processes is challenging mainly due to the fact that it is very difficult for the teachers to observe the students while they apply the process. In addition, technical challenges in student projects often overwhelm the students so they focus on solving these issues and do no longer apply the process correctly (Alégroth et al., 2015; Steghöfer et al., 2016). At the joint Software Engineering Division of Chalmers Technical University (CTH) and the University of Gothenburg (GU), we have therefore changed our approach for our basic software processes courses within our Software Engineering study programmes and for the Software Engineering projects we teach to students outside of computer science. The key change was to adopt workshops in which students apply Scrum to build a city out of Lego (Krivitsky, 2011). This practice is used in industrial trainings and universities (Paasivaara et al., 2014) and has been used successfully for many years at the University of Augsburg.

While we observe positive effects in student engagement and subjectively also in achieving the learning outcomes, we have so far not had reliable data on the actual effects and the perception from the student side. In particular, we were lacking information about how helpful the workshops are when knowledge and skills taught in the workshops are applied in project settings. This paper remedies this situation and reports on our analysis of survey data from different occasions, feedback from student evaluations, and data from student experience reports as well as teachers' perceptions and observations to provide an overview of benefits, limitations, and best practices of teaching an agile Software Engineering process like Scrum using Lego bricks.

In a continuous evolution of how we introduce Scrum, we have applied an action research approach to plan, act and reflect (Dickens and Watkins, 1999) on how to conduct and follow up the Lego Scrum workshop. Throughout the action cycles we have collected different data sets for analysis and validation of our conclusions, based on which we will answer three research questions:

RQ1: What are the benefits and opportunities of using Lego workshops to teach Scrum?

RQ2: What are the trade-offs and limitations of using Lego workshops to teach Scrum?

RQ3: Which of the skills and knowledge from the workshop did the students transfer to their project work?

Since this is a reflection on our own practice, we are going to use Brookfield's four lenses (Brookfield, 1995) to evaluate the workshops from our own *autobiographical* lens as teachers, the *student* lens, a broader *peer* lens by inviting other teachers to observe and participate in the workshops as well as discussing our emerging results, and finally a *theoretical* lens in terms of previous research on teaching Scrum.

Contribution: The main findings of our analysis are that students with low initial confidence feel more confident in applying agile practices after going through the workshops with some high-confidence students actually showing a decrease in confidence. In addition, we find that students often apply Scrum only partially and struggle with task breakdown and estimation. The third and main contribution of this paper is a discussion of the above research questions in the context of an analysis of how the students apply the knowledge from the workshops in actual development projects after the Lego exercise (RQ3). This is a substantial extension of previous work on using serious games to teach Scrum—such as Paasivaara et al. (2014); Lynch et al. (2011) and others—which have not considered the impact on later projects. In addition, we provide concrete guidelines for teachers that are looking for ways to apply Lego Scrum workshops or similar techniques in their courses.

Overview: Our contribution is structured to reflect Brookfield's lenses (1995): The peer lens is found in Section 2, methodology and the setup of the Scrum workshops in Section 3. Section 4 introduces the student lens by drawing from data from surveys, reports, and course evaluations. We then introduce the autobiographical teacher lens in Section 5 and discuss the observations we had in the various instances of the workshop. The theoretical lens illustrates how teaching Scrum with a focus on serious games and Lego is discussed in the literature in Section 6. The four lenses serve as the foundation of our discussion in Section 7. The paper concludes with an outlook to possible future improvements of the workshops and further areas of study.

2. Methodology

In the following, we outline our action research methodology, the context of the application of the Lego exercise and the data collected from each exercise.

2.1. Action research

Action research is used to understand, but most of all improve, real-life situations in an iterative way (Kember and Gow, 1992; Lewin, 1946). The research process is conducted over a series of cycles where each cycle can be broken down into three steps (Dickens and Watkins, 1999): plan, act, and reflect. In the *plan* step, goals are defined and the organisational environment to carry out the change is set up. The plan is then executed in the *act* step, and data is collected that reflects the change. Finally, the impact of the change is evaluated, discussed, analysed, and communicated in the *reflect* step.

A new cycle is initiated as new actions are identified during reflection. The detailed content and duration of each step varies depending on the objectives of the cycle and how the context has changed due to earlier interventions. Stenhouse argues that action research can contribute both to the practice and theory of education and that communicating the resulting insights to other teachers is important in order to promote reflection within the community (cited in Cohen et al. (2000)). The collaborative nature of action research is stressed by Elliot (1991), who states that it is necessary to involve other teachers, a practice Brookfield denotes as the peer lens of reflection (Brookfield, 1995), to fully utilise the process.

2.2. Cycles, Sessions and Data Collection

Our collected personal experiences are based on seven cycles representing the application of the exercise to six different courses, totalling 15 sessions of the workshop. This means that each cycle, corresponding to one course instance, involved one or more exercise sessions. Figure 1 gives an overview of the cycles together with the main data sources. Each cycle is detailed by the program and the term of the course instance, so that *SEM-T1: Software Processes* refers to the Software Engineering and Management program, first term course on software processes. In four cycles we held multiple exercise sessions since we perceived that it would be difficult to handle all the students at the same time. In two cycles, two and five, the exercise was held as an extra-curricular event through invitation by the teacher responsible for the course. The other five cycles were chosen pragmatically – we were course teachers and the courses had learning objectives that we perceived could be met by the Lego exercise. We also state the month and year the exercise took place, the main data source(s), the number of sessions as well as the number of authors present per session. The following section will detail each cycle in order to supply the context and thus enable comparisons between our results and other studies, in accordance to the recommendations for empirical research within both education (Bassey, 1981) and Software Engineering (Dybå et al., 2012).

SEM-T1: Software Processes. The first course we applied the exercise to was *DIT545 Software Processes*, held for the first term (T1) students in the Software Engineering & Management (SEM) bachelor programme of the University of Gothenburg in November 2014. The programme is taught in English to an audience of Swedish and international students. Four sessions were held with 15 to 25 students per session. Since this was the first time the exercise was held in Gothenburg, the groups were kept small to allow a more direct interaction with the students.

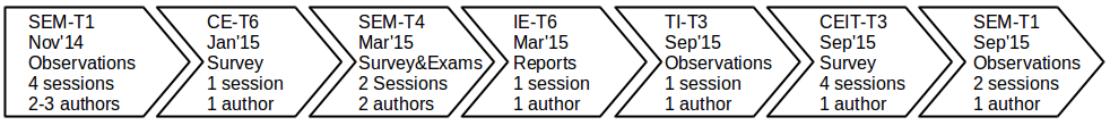

Figure 1: The seven course instances in which the LEGO Workshop was applied.

CE-T6: Extra-curricular Event. The second cycle of the course was given in the context of the Computer Engineering (CE) programme at CTH to students in their sixth term (T6) during January 2015. Some of the students had previously participated in a project course where Scrum was introduced through a lecture and where the students had the task to adopt Scrum to their own needs. From the perspective of the course responsible, the event was extra-curricular, participation was voluntary and the aim was to have an event that reminded the students of the importance of communication and collaboration in project work. The exercise had no relation to course examination. From the authors perspective it was an opportunity to explore what the students remembered from their previous Scrum education. More than 40 students were present during the single session that was conducted in Swedish. The students were asked to fill out a survey[1] in English about their confidence prior to the workshop and answer the same questions after the workshop, thus contributing to the quantitative data collected. A total of 34 replies were received, where six surveys were only partially answered. This is partly explained by the fact that the exercise was given as an extra-curricular event and some students had other courses and obligations which meant that they left early or arrived late. The teacher responsible for the course was present throughout the exercise and shared his impressions with one of the authors after the exercise was finished.

SEM-T4: Software Process Improvement. We furthermore held two sessions of the workshop in March 2015 in the course *DIT548 Software Process Improvement* that is part of the fourth term (T4) of the SEM programme. Again, the sessions were held in English to a mixed group of Swedish and international students. The students all had previous knowledge of Scrum and had applied it in at least one project before. During the two sessions, approximately 45 and 30 students were present, respectively. They were asked to fill in the same survey as the CE-T6 students in the previous cycle and answered the questions before and after the workshop. A total of 72 replies were received. We also used data from the anonymous course evaluation which is an on-line instrument using a five-point Likert scale to assess how the students perceive aspects such as the clarity of the learning objectives, course administration and learning activities. Each question is complimented with a free-text field where the students can write a comment and we collected all comments that mentioned the Lego exercise. In addition, the students were asked to reflect on their experience in a report.This experience was used as the basis for a software process improvement initiative that the students were asked to design and implement in a second workshop. We focus on the reported issues with the process since these emphasise the learning opportunities of our version of the Lego Scrum workshop. Furthermore, they show which aspects of the process students struggled the most with.

IE-T6: Software Engineering Project. The fourth cycle of the Lego workshop was given for students from the Industrial Economics (IE) programme at CTH in March 2015 as part of the course *DAT255 Software Engineering Project.* The course is taken in the sixth term (T6) by predominantly Swedish students with some international students. The language of instruction was English. The workshop had a total of 36 participants, all with no prior experience in software processes or Scrum. The workshop was held at the beginning of the term and students were asked to later apply the process to develop software in the course project and reflect on their application of Scrum in a report at the end of the course. The reflection reports therefore reveal which parts of Scrum were perceived as readily applicable by the students and which parts they struggled with, thus revealing the shortcomings of the Lego Scrum workshops with respect to transferability of knowledge into practice. Examples of what the team were asked to reflect on are roles, agile practices, established routines and lessons learnt. We opted to not use the survey again since at this time we were more interested in gaining an in-depth understanding of how the teams learned and applied Scrum in their course projects.

TI-T3: Extra-curricular Event. The next cycle was carried out at The Hague University of Applied Sciences' campus in Delft. One of the authors was invited to give a seminar about the experiences of the Lego exercise as well as organise one workshop. In this way the emerging theories on how to conduct the workshop, with benefits and shortcomings, were shared and discussed with peers outside our own faculties. 24 students from the Technical Informatics (TI) programme participated and the session was given as an extra-curricular event. The students were in their third term (T3) of a bachelor programme and had previously used Scrum in two projects. The course responsible saw the exercise as an extra-curricular event, meaning participation was voluntary and the exercise had no relation to course examination or learning objectives. From the authors perspective it was an opportunity to explore

[1] https://gubox.box.com/v/2017-JSS-No-Silver-Brick

how the students applied their previous Scrum education in a new setting. The main data source from the exercise was observations.

CEIT-T3: Software Engineering Project Revisited. The sixth cycle was conducted as a part of the Software Engineering Project course. The difference from the spring version is that in the autumn the course is taken by third-term students from the Computer Engineering and Information Technology (CEIT) programmes at CTH. Four different sessions were conducted with a total of more than 130 students. After each session the students answered the same survey questions regarding confidence that were applied for the CE-T6 and SEM-T4 students. Since the students had no prior experience of Scrum, data about their experience before the workshop would not have been useful. A total of 131 responses were collected after the workshop and 87 students answered an additional survey with the same question set at the end of the course. The survey and observations were the main data sources.

SEM-T1: Software Processes Revisited. Finally, the last cycle was given to the SEM-T1 students of 2015, closing the cycle. The exercise was conducted in two sessions with 45 and 35 students respectively. We used observations as the data source to validate our conclusions from the previous sessions. The exercise was conducted earlier this year to better align the learning outcomes with the parallel programming project.

Summary of Collected Data. In total the Lego workshop has been applied to six different courses and seven course instances, resulting in 15 sessions which have generated nine reflection reports, 100 survey answers comparing the confidence before and after participating in the exercise, 131 survey answers regarding the confidence after the workshop, 87 responses accounting for the change in confidence of applying Scrum in a course project as well as 70 written exams. Approximately 450 students have been observed while participating in the exercise. The exact number is not possible to provide since participation in the exercise was always on a voluntary basis (the exception being the SEM-T4 students who had to reflect on the exercise to pass the course), which means that some students left early or arrived late. The observations were conducted openly, meaning that the students knew that they were being observed, and documented as field notes and these notes were later on used for analysing the Lego exercise in relation to our research questions.

The sprint reviews and retrospectives are also an opportunity for informal interviews to collect data about issues that we observed repeatedly. In this way we could describe an observation for the students and then ask for their interpretation of the data. The explanation could then be discussed among the authors but also presented to students in later sessions to get their response.

2.3. Data Analysis

When we started out with the first Lego Scrum session we did not know what to expect from the students, which the teachable moments were, or what would trigger them. In short, we did not know what to expect in terms of opportunities and limitations besides introducing Scrum in a new way to our students. Subsequently we began with an exploratory view of the exercise and an *inductive analysis* (Runeson et al., 2012) of the free-text data. This means that we treated the reflection reports, the field notes and free-text comments in the same way during the qualitative analysis.

The first two cycles were mainly used to see what was possible and what kind of interaction we could have with the students during the exercise. This was done by the three first authors individually. The first step was to label occasions where the students deviated from the instructions or expectations but also for examples of causality where practice had a clear impact on how successfully the teams managed to carry out a task, a common way to conduct and analyse field observations (DeWalt and DeWalt, 2002). From these labels we then collectively formed themes regarding the opportunities and shortcomings for learning Scrum through the Lego exercise.

As we formed more and more themes we shifted to a more *deductive analysis* (Runeson et al., 2012) as we saw which themes were commonly repeated over the sessions and cycles. We could also start to look for explanations for why the students acted as they did during the exercise by asking them. This enabled us to consolidate our findings, at which point we investigated them in relation to existing literature to further understand the implications for learning Scrum through the Lego exercise.

The survey data was analysed using descriptive statistics to calculate the mean and standard deviation for each question. In the case of the responses collected from the CE-T6 and SEM-T4 students, we also calculated the difference for each student in terms of change in confidence before and after the exercise. We also grouped the students into three different experience levels in order to assess how prior experience of Scrum affects the learning opportunities of the workshop. When a student qualified for one or more experience levels we took the highest possible experience level.

3. Setup of Scrum Workshops

Scrum is a project management framework for development projects that is based on the principles of cycles, small self-governing teams, and a customer representative called *Product Owner* (PO) (Schwaber, 1997). Scrum is often but not exclusively used to manage software development projects. The iterations allow continuous refinement of the developed product, where the end of each iteration (called a *Sprint*) allows the customer to give feedback during a *Sprint Review* meeting. The team members can

reflect on their performance and discuss improvements of their way of working in a *Sprint Retrospective*. Requirements are usually collected as user stories on a *Backlog*, an ordered list that is physically available to the *Scrum Team*. The PO writes the user stories and prioritises them by their perceived value. At the beginning of the Sprint, the Scrum Team conducts *Sprint Planning* during which it estimates the effort required for the user stories, potentially engaging the PO for clarification, and selects as many stories as fit into its *Velocity*. A team's velocity is continually determined as an empirical value which reflects the team's past performance.

3.1. Scrum Workshop Goals

At CTH and GU, Scrum has traditionally been taught as part of project courses during which the students are encouraged to apply Scrum. However, the process is not in the focus of attention in these courses and we have observed that the focus on the product and the technical issues associated with it distract students from the process (Steghöfer et al., 2016). The intended learning outcomes with respect to process knowledge and ability to apply it are therefore not achieved. Inspired by the successful use of Lego Scrum workshops (Krivitsky, 2011) at the University of Augsburg, we have therefore adopted the format of the simulation for use in Gothenburg. In doing so, we pursue the following goals:

- Introduce the process in a way that puts the process in focus rather than the product.
- Have multiple iterations (sprints) in sequence so that process improvements discussed in the Sprint Retrospective can be acted on and evaluated, forming a sequence of plan-act-reflect.
- Inspire the students to take initiative, be proactive, and take responsibility for their own work.
- Give teachers the opportunity to observe, evaluate, and comment on the use of the process directly.
- Prepare students to work in the industry by providing experiences similar to those in the industry.

These goals are in line with the intended learning outcomes of the courses where the workshop was conducted.

A major advantage is that students will not only be able to describe these aspects as presented to them in a theoretical fashion in a lecture, but will be able to discuss them based on their own practical experience, allowing theory to resonate with practice (Kolb, 2014). By introducing stakeholders who act in a non-optimal fashion, by encouraging cooperation between teams, and by letting the students take initiative, we simulate conditions in the industry.

All workshop instances follow a plan very similar to the one sketched below, with changes only to accommodate different time slots (e.g., with a lunch break) or to react to learning opportunities that arise during the workshop (see Section 7 for more information on this). The overall setup allows us to align the learning objectives for the workshop with our overall goals as described above. Concretely, the setup of the workshops intends to create an environment which encourages communication with stakeholders, encourages the plan-act-reflect feedback loop (Dickens and Watkins, 1999) and allows going through it multiple times, as well as exposes the students to stress to provoke un-reflected behaviour and straying from the process and thus create learning opportunities.

For all courses regarded in this paper, including the one in Augsburg discussed later on, process knowledge used to be mostly conveyed through lectures, constituting a predominantly theoretical approach. The workshop format is our attempt to remedy this situation and to give the students the opportunity to experience Scrum in a more practical format.

All authors of this paper have a good working knowledge of Scrum, both from a theoretical perspective as well as from a practical perspective. One of the authors of this paper is a certified Scrum coach, another one has worked with Scrum in industrial settings, and three of them have done empirical research on agile methods. However, an in-depth knowledge of all aspects of Scrum is not necessary to run the workshop successfully. A working knowledge of Scrum, a good command of pedagogical methods, and some experience in running medium- to large-scale workshops with students are much more crucial to setup, run, and monitor the workshops. The workshops provide a controlled setting in which the teachers can discuss "textbook" Scrum. There is also a learning curve for the teachers involved that can help them understand the intricacies of Scrum better and thus convey the process to the students in a better way.

3.2. Introducing the Students to the Workshop

The workshops are conducted by one teacher, often with the help of a colleague. Ideally, both teachers are experts in Scrum, although this is not strictly necessary depending on how the roles are divided. A time frame of approximately four hours is set aside for the workshop. Students are usually not told what to expect in an attempt to conduct the workshop without any bias. Before the students begin their work, an introduction is given in which students are informed about the goal of the workshop, i.e., to cooperatively build a Lego city by showing them a vision (a photograph of a professionally built Lego city from a catalogue). Furthermore, a very brief introduction to Scrum is provided, outlining the main roles and the concepts of the Sprint, the Sprint Reviews, and the Sprint Retrospectives. Students are encouraged to ask questions for clarification and the teachers ask the class whether everything has been understood.

The students then see the prepared Backlog for the first time. A number of user stories are posted on a whiteboard. The teachers explain that they are ordered by priority and

that the stories are prioritized in descending order, so the top-most are more important to the customer than those lower down. One round of planning poker (Moløkken-Østvold et al., 2008) is conducted with five randomly selected students while the others observe. All students receive planning poker cards. One user story is picked from the board and the students are asked to make a first estimate. The students show their estimation cards and the outliers are asked to share their reasoning. The process is repeated until convergence to one estimate is achieved.

Next, the students are asked to form groups of five to eight people and elect a *Scrum Master* for each group. The purpose of the Scrum Master (communicate with the PO and other groups, remove obstacles, conduct the Sprint Retrospective, manage changes to the way of working and monitor the process lifecycle) was explained in the brief introduction to Scrum. The resources are introduced by pointing to the Lego bricks that are still in a container (either a duffel bag or a plastic or cardboard box) and placed in a remote corner of the room.

For an initial estimation of the user stories, the students are asked to take a picture of the current backlog and conduct the planning poker for all user stories. A time span of 20 minutes is set aside for that. Students are asked to write their estimates on post-it notes and place them next to the corresponding user story on the whiteboard. The posted estimates are the basis of a final discussion about the difference in estimates and how each team will set their own velocity. In this context, a brief discussion of the *Definition of Done* is started, emphasising that it is important that each team have clear criteria that describe when their product is ready for review. Finally, the integration area is pointed out and it is made clear that only the products that are in place in time will be reviewed.

3.3. Repeated Sprints

After a final opportunity for questions, the introduction part is concluded and the first Sprint begins. A clock is set for 18 minutes and prominently displayed in the classroom. When the time runs out, a noise will announce the end of the sprint. While the students have been informed about how a sprint is supposed to progress, no further instructions are given at this point. The teachers are available in their roles as Product Owners and will answer any questions the students might have about the product they are supposed to build. They do not interfere with the work during the first sprint.

At the end of the sprint, the students and the teachers gather around the integration area. Each Scrum Master is asked to report on the product that was developed. The teachers in their capacity as Product Owners make statements about the quality of the delivered products and reject or accept what has been built. During the first sprint, interaction with the Product Owners is very rare and the models are usually colourful, of different scale and do not fit together (e.g., in the case of a tractor and a garage for it). After the review, the groups have five minutes to conduct their Sprint Retrospective in which they discuss how to improve their performance. When the time is up, the next sprint starts.

We aim for a total of four sprints, usually with a break after either the first or the second sprint. Each sprint has the same duration of 18 minutes. The Product Owners rearrange the backlog between sprints and add new user stories. The groups must estimate the stories during the sprints. While the students are left unperturbed during the first sprint, some additional hurdles are introduced by the teachers in later sprints. A "sudden server outage" can force an entire team to stop working for a short period of time. A "sickness or accident" can occur to a student, and s/he is removed from the team for some minutes. Other interference in the form of sudden changes in the requirements is also enacted.

The last sprint is dedicated to allowing the students to finalise the Lego city. In the end, they should have had a rewarding and memorable experience and delivering a beautiful product is a big part of this. After the last Sprint Review has been held and the students have been lauded for their work, a brief feedback session is held in which the students can voice what they liked and what they disliked about the event. The teachers give a summary of the most important lessons learned during the workshop.

It is important to note that the teachers also point out issues with the process they have encountered during the Sprint Review. A typical remark after the first sprint is that none of the students engaged the Product Owners to clarify user stories or gather complementary information. While some feedback is given to individual students or groups of them during the sprints, sharing these thoughts during the retrospective allows the teacher to relate theory to practice and make the relationship between student behaviour and outcome immediate.

4. Student Lens from Course Evaluations, Surveys, and Experience Reports

In this section, we use the student lens to give insight into the students' perception and feedback of the effectiveness of the workshops. While we analyse the data here, a discussion of our findings and conclusions that also incorporates the autobiographical lens (cf. Section 5) and the theoretical lens (cf. Section 6) can be found in Section 7.

4.1. Quantitative Analysis of Student Surveys

The survey data from cycle two and three was analysed both across statements to see the change of confidence in the students as a group as well as over individuals. The quantitative data shows that both the CE-T6 and SEM-T4 students have a significant increase in confidence in applying Scrum. However, especially the experienced group includes some students with decreased confidence for some

statements. Finally, the survey data showed that the students perceived the process, rather than the product, to be in focus during the exercise.

In order to derive statements about changes in confidence from our data, we categorised students according to their prior experience with Scrum:

Low experience Students with no prior experience, or who had heard a lecture on Scrum, or who had participated in industrial training. In total 14 students, of which 13 came from the CE-T6 workshop and one from the SEM-T4 occasion.

Medium experience Students who had participated in a Scrum workshop or in a single project where Scrum was used. In total 19 students, of which 15 came from the CE-T6 workshop and four from the SEM-T4 occasion.

High experience Students who had participated in two or more projects where Scrum was used. In total 67 students, all of which came from the SEM-T4 occasion.

4.1.1. Positive Change of Group Confidence

When the impact of the Lego exercise on the students' self-assessed confidence is analysed across experience levels, we found that the final confidence for each survey statement is similar regardless of the initial confidence, see Figure 2. Here the students with low experience end up in the range 3.29 to 4.00 on a five-point Likert scale across the survey statements. The students with medium experience report a mean self-assessed final confidence in the range 3.47 to 4.13 and the highly experienced students lie between 3.52 and 4.23. The impact of the Lego exercise, the intervention, in terms of percentage is given in Table 1. Due to the small populations for the low and medium experienced groups we do not supply any further statistical analysis of the data sets.

Our data implies that those students with lowest experience gain the most in confidence in relative terms while those with the highest experience level gain the highest absolute confidence, see Table 1. So a student with low initial confidence will likely gain more in relation to the initial estimate than an experienced student who will estimate her/his confidence with a higher final value. The data also supports our division of the students into three sets based on experience. The students labelled as having low experience estimate an initial confidence in the interval 2.29 to 3.07 on a five-point Likert scale, the medium-experienced students have mean initial confidence values of 2.67 to 3.33, while the highly experienced students report initial confidence values between 2.97 and 3.71.

The statement regarding estimation of effort and velocity has the greatest gain in self-assessed confidence for both the medium and the high experienced students while the corresponding statement for the students with low experience concerns how to conduct a retrospective. When it comes to the lowest gain, the students with low and high experience share the perception that the usage of new technology is least straight-forward while communication with other stakeholders has the lowest gain for the students with medium experience.

4.1.2. Negative Change of Individual Confidence

While the students at large gain in experience for all statements, some individual students report lower confidence after the exercise than before. The distribution of individuals with a negative change in confidence is given in Table 1 along with experience level and survey statement.

It is worth noting how the number of students with a negative confidence change drops as the number of negative changes increases. So, whereas 23 out of 100 students report that they have lower confidence for one survey statement, only two students report a lower confidence for four survey statements. There were no students who reported lower confidence for five or more survey statements. In total 39 out of 100 students reported that they had lower confidence after the exercise in relation to one or more questions. Distributed over experience level the result is 2 out of 14 students with low experience, 8 out of 19 students with medium experience and 29 out of 67 students with high experience. A point of interest here is that one of the students who reported three negative changes in confidence had only theoretical knowledge of Scrum from reading a book and had never applied Scrum prior the workshop. 23 students that reported a lower confidence did so for only one question, ten students reported lower confidence for two questions, four students for three questions and two students for four questions.

4.1.3. Process over Product

Besides asking for the students' prior experience of Scrum and how confident they were in applying the basic Scrum practices, we also wanted to see what they perceived as the main goal of their last project. The outcome is depicted in Table 2. For the students with a high experience of Scrum, 43 individuals stated that they perceived that product was in focus during their last project which dropped to 17 for the Lego exercise. Similarly, the number of medium experienced students who saw process as the focal object rose from eight to eleven when comparing the last project to the Lego exercise. We excluded the responses from the students with low experience since they by definition had not applied Scrum before the exercise.

4.1.4. Project Impact on Confidence

From the survey data collected from cycle six and the CEIT-T3 students we have 131 responses after the Lego workshop and 87 responses after the course project. While we cannot follow individual changes across the surveys since they were anonymous, we can compare how the population has changed over time as the students have had the

Table 1: Confidence change with regards to experience level and statement and number of negative confidence changes

	Experience						
Survey Statement	Low	# neg	Medium	# neg	High	# neg	Total
Applying Scrum in next project	+37%		+29%		+13%	6	6
Estimating effort and velocity	+38%	1	+39%	1	+27%	4	6
Breaking user stories into tasks	+32%		+22%	1	+12%	8	9
Communicating with stakeholders	+30%		+16%	3	+11%	10	13
Using relevant technology	+24%	2	+20%	1	+6%	7	10
Conducting a review	+44%	1	+33%	1	+12%	3	5
Conducting a retrospective	+50%		+35%	3	+16%	1	4
Meeting deadlines	+37%		+18%	1	+8%	9	10
# Respondents		2/14		8/19		29/67	39/100

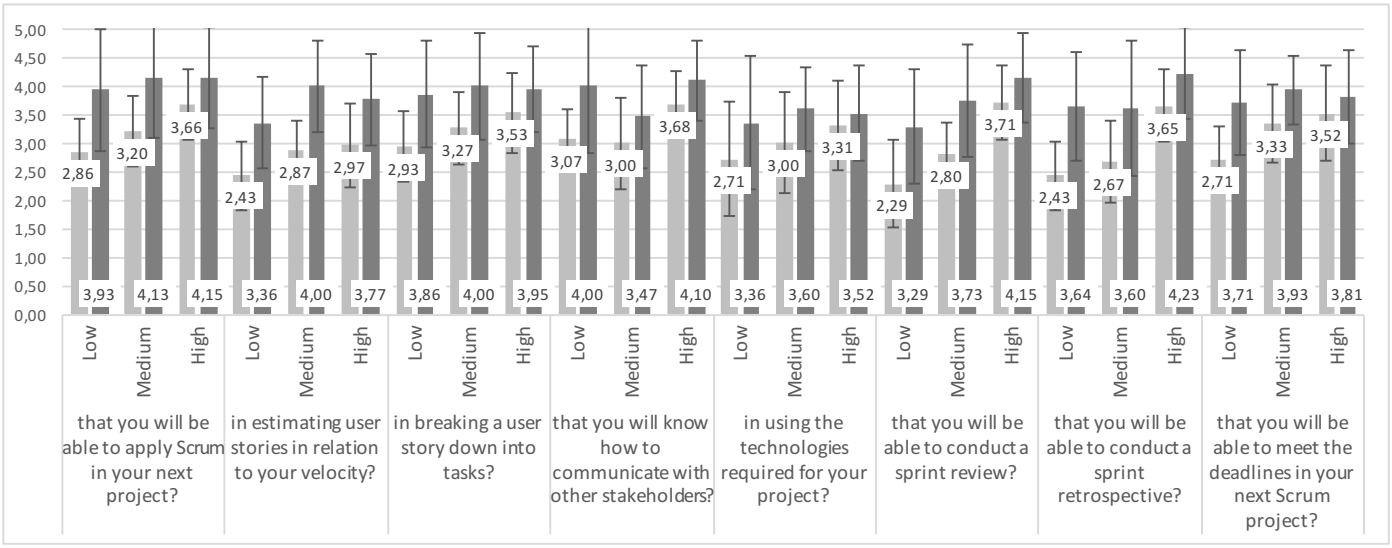

Figure 2: Changes in confidence for students of different experience levels. Light gray staples represent confidence prior the exercise, dark gray staples represent confidence after the exercise. Each statement is the continuation of the phrase *"How confident are you . . . "*

Table 2: Response rates regarding if the process or the product was in focus during the last Scrum project

	Last Project		Lego Exercise	
Experience	Product	Process	Product	Process
Medium	8	8	6	11
High	43	23	17	35

Table 3: Changes in confidence from after the workshop to after the project

	Confidence after...	
Survey Statement	workshop	project
Applying Scrum in next project	3.7	3.8
Estimating effort and velocity	2.9	2.9
Breaking user stories into tasks	3.7	3.7
Communicating with stakeholders	3.6	3.3
Using relevant technology	3.6	4.0
Conducting a review	3.6	3.7
Conducting a retrospective	3.7	3.6
Meeting deadlines	3.4	3.3

opportunity to apply what they learned during the exercise to their course project. The mean values for each confidence statement are given in Table 3. The only noticeable changes regard the drop in confidence for communicating with external stakeholders and the increase in confidence regarding technologies.

4.2. Qualitative Analysis of Student Reflection Reports

We use two main sources of qualitative data for this analysis: The IE-T6 and SEM-T4 reflection reports described in section 2.2. In addition, we use some information from the SEM-T4 course evaluation and from the surveys we conducted in the different courses. The concrete source of the data is mentioned whenever appropriate.

4.2.1. Opportunities

The students' reflection in their experience reports for SEM-T4 was centred around which problems they observed

during the Lego Scrum workshops. Later on in the course they were asked how they would address these issues. Here, we only regard their reports on the problems, however, since these are indicative of the learning opportunities created during the workshops and show which aspects of the process are surprising to the students. One student summarises his experience as follows: *"A lack of understanding of the process is to blame for most issues rather than the process itself."*

One major theme in the reports is *communication*. Discussions of three different kinds of communication appear time and time again: communication within the team; communication with other teams; and communication with the PO. *"The poor communication within the team and with other teams had a clear impact on the end-products."* states one student. Likewise, another student reports: *"Even though workshop's goal in the end was to build a city together, teams did not communicate as much and it felt more as a competition. We did not communicate with product owner too which only resulted in defected deliverables."* The students pick up on the fact that communication with the PO is necessary after the first Sprint Review sees most products rejected. However, the right way to discuss issues with the PO is not obvious: *"Our questions to the product owner were at sometimes little bit unclear; this caused some confusion during assembly when we recognise that we needed more details in order to continue. An example of this was: 'What colours would you like?' The product owner was then free to choose colours that were not available. For the limited options of colours to the colours we had [sic], we wasted time to go back and reorganise the colour options with the product owner."* Issues with communication between the teams often became evident during the integration stage: *"Moreover, since we were not the sole Scrum team working in the same context, our work often involved integration with products by different teams. Early enough, we observed that this was not something trivial, since we often delivered artefacts that were not compatible with those of the other team."* Within the teams, communication was also limited due to the fact that many teams chose to split the work between team members in a way that allowed them to work alone. We interfered with the teams by applying the "truck factor", i.e., taking a member out of the team to demonstrate that the project is in danger if one of the team members is hit by the metaphorical truck: *"It was also discovered that time was wasted on trivial tasks, such as drawing unused or unusable design documents. [...] Sprint three found that some sort of design documentation is necessary after the truck-factor came into affect [sic]."* This illustrates that the students realised the value of documentation as a communication medium.

A further repeated theme are the *vague requirements* that the students were provided with. The requirements were, of course, intentionally bad to ensure that the students realise that they must not make assumptions about the requirements but rather find out what the PO actually has in mind. One student remarks *"The requests from the stakeholders were too vague on the product backlog and we had guessed on many things."* while another says *"They [the product owners] also often deliberately presented us with vague or ill-defined requirements yet with occasionally strong ideas about what was and was not acceptable."*

Many teams had issues with *knowledge and resource acquisition*. Both aspects present a significant effort both in project courses (Steghöfer et al., 2016) as well as in real-life projects, mostly with a focus on knowledge acquisition. In all project courses in the Software Engineering and Management bachelor programme, students are asked to apply new programming languages, frameworks, or tools. Even though the students in the SEM-T4 course knew that, they still did not take the effort for this into account. However, this caused problems later on: *"[Ne]ither the builders or the allocators knew from the beginning which Lego parts that were needed in order to finish the user requirements, that made problems in the middle of the building process when we realized that we need a particular part."* One student complained about *"not even seeing the Lego before starting"*. It must be emphasised, however, that we did not enforce this. We told all students where the Lego can be found. In those workshop instances in which students ask to see the Lego, we happily oblige them. If students do not ask, however, we do not tell them about the possibility. Likewise the search for resources (i.e., Lego bricks) is mostly not taken into account when doing task estimations. One student reports: *"Some of the problems we faced were setting a too high velocity for our first sprint hence overestimating our capacity, one part that caused this was that we did not realize that allocating parts took much longer than anticipated."*

One particularly interesting source of learning opportunities is the incorrect or partial application of Scrum (Babb et al., 2013). The workshops are especially well suited to reveal such problems as the teachers are able to observe the students and address these issues as soon as they come up. One of the problems we repeatedly observe was picked up by one of the students as well: *"When the sprint started, the rush for picking stories and pieces of Lego was stressful. Already at this stage the groups didn't coordinate which stories is highest prioritised."* We do, however, emphasise that the order of stories in the backlog represents the perceived value and that we expect to have the most valuable stories delivered first. Similarly, while we introduce the definition of done briefly, we let the students realise its meaning and purpose by themselves: *"We didn't know what 'the definition of done' really meant until we experienced it."* The purpose of the Scrum Master and its role for and in the team is also often difficult to understand for students when they are trying to make ends meet during the sprint: *"Whenever something went slow, or a small problem occurred I [as the Scrum Master] always stepped in myself. Instead of reflecting over the problem, its cause and trying to solve the core issue together with the team, I continuously ran people over with*

my duct tape. Not only is a process where I run around (and let me exaggerate) fixing everything, a very unstable and in-deterministic process, but it's also very hurtful for process improvement. Not only does it become harder for my colleagues to recognize problems because they are constantly removed before they recognize them, but the fact that I personally took on so many made it impossible for me to remember them."

Finally, a particularly difficult part for the students is *task estimation and breakdown. "[. . .] when it was time to work, everyone just started working, before actually knowing who was going to do what and how, this was going to result in some negative things. In the end of the sprint, the quality of the product was not so good and there were group members that had nothing to do."* We frequently observe similar problems in the first sprints. In a way, a missing breakdown of tasks and distribution to team members is also a form of lack of communication within the team. Many students realise that some planning in the beginning is valuable, even under the stress of the 18 minute sprints: *"An extra minute or so at the beginning of the sprint to just assign members to particular roles could've deterred a lot of confusion."*

4.2.2. Uniqueness of the Workshop in the Curriculum

One additional observation that is of note comes from the official course evaluation of the SEM-T4 course. There, one student points out one speciality of the workshops that is unique in this setting and does not appear in the rest of the education, particularly not in the project courses that follow in the SEM-program. The student states: *"During this course was the first time I faced changes in requirements from the product owner/stakeholder and also the first time that the scrum team needed to be shielded by the scrum master from this person."*

This shows that the workshops offer a benefit as compared to project courses. These usually require one of the team members to act as the PO (similar to the way it is handled in Werner et al. (2012)). Naturally, this team member will not act in the way an actual PO might, changing their mind, trying to shift the scope mid-sprint, etc. Therefore, the Lego Workshops expose the students to a setting that is realistic and that can not be achieved elsewhere in their education. A similar notion is taken up in one of the experience reports when a student mentions this uniqueness of the workshop setting: *"The seemingly obvious idea of improving requirements by just talking to the product owner is possibly missed because the environment they normally work in is one of being given assignments and being expected to just complete the work."*

Another unique aspect of the Lego Scrum workshops is the fact that students need to work together to create the final product. Usually, the project courses in which Scrum is applied enforce that each team develops its own product in isolation from the others. This also causes a certain mindset for students that have experiences with such courses: *"The lack of communication between teams can be contributed to the fact that we so far only have worked in separate scrum teams without having to share resources with other team or communicate about size or standard [. . .]"*

4.2.3. Application of Scrum in a Subsequent Project

While the comments discussed so far all stem from a reflection on the Lego Scrum workshops itself, we are now going to move to evidence on how the students apply the learned knowledge in a software development project that spans several weeks. The data used below is from the IE-T6 students' reflection reports in which they reflect on how they have used Scrum in the project. In particular, the results highlight which parts of Scrum have not been adopted by the students and thus give an indication of the shortcomings and limitations of the workshop format. They also allow to answer RQ3 which concerns the transfer of knowledge and skills into the project.

Interestingly, many of the issues that we have identified in the feedback on the workshops also show up in a software project setting (Steghöfer et al., 2016). Predominant problems reported on are issues with task estimation and breakdown, incomplete application of the process, underestimation of knowledge acquisition, and a learning curve that surprises the students. In general, however, none of the student groups reported that they felt Scrum did not fit the project. On the contrary, almost all groups explicitly state that the process helped them. Especially the structure provided was seen positively: *"It also feels that the group's work with the new approach SCRUM gives all members new tools to structure a project."* However, none of the groups reported on applying a "pure" process, but rather all adapted Scrum to some degree. This has been identified as an issue in the literature (see, e.g., Babb et al. (2013)). In the following, we will present qualitative data for the aspects mentioned above to show how students apply their new skills and knowledge in a project.

A number of teams reported on problems with *knowledge acquisition* at the beginning of the project. The technical knowledge that needed to be acquired as well as the domain knowledge proved to be more challenging than students anticipated. As one team reports, *"we thought it would be a simple task to learn to use GitHub, which ultimately took much longer than we had expected."* The lack of knowledge also has an impact on the ability to estimate and plan. If knowledge acquisition is neglected in the beginning, this can cause problems later on. *"We would probably have been able to avoid having to do things that were not planned in the sprint by making a more proper planning and really think through how everything would be integrated from the start, but this would clearly have been much more time consuming and also requires more knowledge on Android that we did not have in the beginning of the project."* However, it is not only knowledge that is required, but also experience. Therefore, estimates are expected to be slightly off in the beginning of a project but also expected to become better. This is acknowledged by

the students as well: *"Probably, the fact that the estimate of the time required has become better and easier has to do with more experience and greater knowledge of programming but some uncertainty will probably always remain."*

Issues with *task breakdown and estimation* can, however, be identified in the reports of almost all groups. One team reports: *"the application of Scrum and creating features in parallel was difficult to manage. This was due to the fact that basic functions such as the database connection had to be done before it was possible to divide the work."* This indicates that the team had issues creating vertical slices (Cockburn, 2010). The same team reports that *"sprints were considered at times to be somewhat short"*, an indication that the effort of the selected tasks did not fit the velocity or that the velocity was overestimated. At the same time, the dependencies introduced by this kind of task breakdown make it hard for the team to distribute the work amongst them: *"The disadvantage of this type of planning process was that the workload during certain weeks were unevenly distributed between the project team members."* While some of the groups tried to apply planning poker, some abandoned this technique altogether and used a more ad-hoc approach: *"No systematic approach to estimate our user stories effort was used, however. The decisions were based on experience. This meant that the estimate was difficult to do at first but became easier during the project."* Others, however, struggled with the estimations: *"We had no real idea of the complexity of our tasks and no good way to determine which effort a particular task demanded."* Related to task breakdown and estimation, some of the groups report that *planning* in general was not helpful for them. *"In Sprint 2, we tried to work with tasks, which was more cumbersome than rewarding. The cooperation of the group was so good that clear user stories sufficed."* The time taken for planning also felt disproportionate for one of the teams: *"Moreover we realized after a while that planning in Scrum took a lot of time in relation to how much time we actually spent on the app itself."* Others found it hard to adhere to the plans that have been set: *"Sometimes it was difficult to stick to the plan for each sprint and it happened that we did things that were not in the sprint planning."*

All reports show that Scrum has been to some extent *adapted* by the teams or at least only been *adopted partially*. One team, e.g., reports that *"we did not appoint a ScrumMaster because we did not really understand the meaning and importance of the role"*. This particular team also left out other aspects that were part of the workshop. About the sprint retrospective, the team says *"In retrospect, we realize that we missed an important aspect of these meetings: we never really evaluated a previous sprint, but looked only forward."* This caused problems with their effort estimation. The team's explanation for leaving out certain elements was that the planning took up too much time.

Another common omission was the differentiation of the Product Backlog and the Sprint Backlog. One team reports that they only had a backlog for the current sprint and discuss the impact of this: *"In a similar project in the future we would develop the sprint planning to also cover a little more long-term perspective. We felt that our plans that covered only a week were a bit too detailed to get an overall picture of the five-week project."* Another issue that was inherent in the project setup is that there was no dedicated PO. While some teams assumed the PO role themselves, others counted on the supervisor as the PO, a role the supervisor did not officially fulfil, though.

A number of *misunderstandings of Scrum* were evident. One group reports: *"We learned from the Lego exercise that we shouldn't take the more demanding features to start with since it's better to have something done completely rather than just nearly."* We emphasised in the workshops that those stories that deliver most value to the customer are the ones that should be delivered first. In an actual development project where the development team negotiates the priorities with a PO, the initial prioritisation can also be based on risk or expected impact. The students perception of "demanding features" is at odds with the principles communicated by us. Another group reports that *"we went sometimes away from the planning, but in those cases it was things we discovered that needed to be fixed before the rest could be implemented."* We discuss the possibility to add stories for fixes to the backlog during the workshops, but none of the teams picked up on this.

Interestingly, many teams report of a *learning curve* for process adoption. *"It was also a learning process to begin working with the SCRUM approach, which the group took for granted that they would be able from the outset, but that turned out to have a certain learning curve."* reports one of the teams, while another writes *"To first understand how we should work with Scrum, however, took some time and continued to develop gradually in the project."* Many report a learning curve for sensible estimation as well, as witnessed by the comments cited above.

Other notable factors that were, however, only mentioned by one of the teams are, e.g., the *need for refactoring*: *"[...] our programming at times became shortsighted and only solved the problem to be solved for the moment. Thus, we not always developed code that was optimal for subsequent implementations."* Another team reports problems with the iterative approach used in Scrum: *"Tendencies to some frustration have arisen among project members, because of unfamiliarity using ongoing and frequent documentation and planning."* Other teams find this to be one of the major strengths of the methodology: *"The biggest advantage that the group identified for using SCRUM was that after the first sprint an application that could be tested and built upon was available."* reports one team while another writes *"By continuously planning, executing and evaluating, a better result is probably achieved than would have been possible with a waterfall method."*

5. Autobiographical Teacher Lens from Observations

While many of the observations from the students were also picked up by us as the teachers — such as the problems with task estimation and breakdown — there are a number of additional issues that we can see. The following list is not intended to be complete. Rather, it details the main aspects that consistently show up in the workshops.

More Scrum experience doesn't make the exercise easier. We have conducted the exercise with students who have already been exposed to Scrum in previous project courses but also students without any prior knowledge. Interestingly, the problems that came up were the same. The students with prior experience did not do any better in terms of communication, task breakdown, planning, or collaboration. The reasons for this are still subject of discussion. Possibly, the students applied Scrum in their previous projects without consideration of these aspects. An additional possibility is that Lego is sufficiently different from the software students have built in prior experiences that they fail to transfer their skills to the new medium. However, our observations show that students do not even attempt to break down tasks, e.g., so it is doubtful if that is the main issue.

Students do not think outside the box. Students tend to stick to a very limited set of tools. One of the user stories we introduce in the later sprints is to have roads that connect the different buildings. Even though the integration area is made of paper and there are markers lying around everywhere, students tend to try and use Lego pieces to mark the roads. Similarly, the user story for the radio tower calls for a satellite dish, a building block not easily assembled with the available pieces. In both these cases, students fail to see that they have options apart from the provided Lego. A partial explanation is that the students get to respect the challenges of finding the right building blocks and understanding what is possible and what is not in terms of Lego. This in turn leads to a growing confidence in relation to the Lego, and under stress the Lego becomes a silver bullet (Brooks, 1987). Paraphrasing the saying about hammers and nails by Kaplan (1964), the students tend to turn all user stories into challenges that can be handled with interconnected plastic blocks.

This problem might be linked to how the game is explained in the beginning. In Augsburg, the Lego Scrum workshop is usually introduced as "[L]earning Scrum by doing Scrum in a real development project, but to remove the technical obstacles resulting from unknown frameworks, APIs and programming languages required to do a software development project. Therefore we use something that everyone can use even without prior introduction: Lego." This might lead to some students taking it for granted that they are freed of thinking about alternatives or building material. In Gothenburg, the introduction lays more focus on building a Lego city. Putting the material on the forefront might make it harder for students to take the mental leap towards other building material.

Failure to define what "Done" means. We engage the students in a discussion of their Definition of Done before the first sprint and encourage them to define within their team what "Done" (or rather "Done, Done, Done") means in the context of the Lego exercise. Interestingly, the students usually dismiss this request and proceed without a reasonable definition. Once the first chaotically multi-coloured model is presented during the sprint review, we ask if they consider the model "done". At this point, the students realise that the Product Owners have a specific notion of "done" not explicitly included in the acceptance criteria. This notion includes a coherent colour scheme. However, the definition of done remains underspecified for most teams throughout the entire exercise.

Failure to break user stories down into tasks. The students consistently have a hard time splitting the user stories into tasks. If they attempt it at all — which is rare enough — they usually abandon the practice after the first sprint when they realise how much work they chose to take on within a sprint. That causes problems, especially in later sprints when we remove students due to a simulated "sickness". In these cases, the work that was done by the removed team member often is merely picked up by those remaining without real knowledge what was supposed to be done. We observed instances where half-finished models were disassembled again since the remaining team members did not understand the original intentions. Having a task breakdown for the model and being able to see which tasks are left and which already have been completed would arguably lead to a smoother replacement of missing members.

In Augsburg, a slightly different situation can be observed. The students do indeed break down the user stories into tasks, but the tasks are not used to track progress. Once they are broken down, they are placed in the "to do" column of the Sprint Backlog, but are never moved to the "in progress" and subsequent columns. That the practice is applied at all is most likely due to our introduction of Scrum in lecture form before the workshop. In the Scrum lecture, we emphasise the activities and expected results of the Sprint Planning meeting, especially a properly prepared task board. When asked why the task board has not changed at the end of the sprint compared to the start of the sprint the students gave different answers:

- Time pressure is high in the sprint and everyone, even the Scrum Master, is investing all the available time in building. This is an example of the participants forgetting about the process and solely focusing on the product. The inexperienced Scrum Masters add to this problem.

- The participants do not see any benefit in updating the task board. They often organize their work in the actual sprint in an one-person-doing-one-user-story way. The tasks are often a linear decomposition of the required building (find pieces, build first floor, build second floor, put floors together, add roof, ...). The person doing the building knows perfectly where she is in this sequence, so having this information on the task board does not add value. This only changes when we start removing team members due to "sickness".

- Some teams report that it is faster to ask another team member to learn what the current state of the user story is and what is to be done next rather than to look for this information on the task board. The reason for this is that the Sprint Backlogs of some teams are not located where the building takes place. Some teams even keep their task board in a different room, a failure of the teams in properly preparing their workspace.

Overfitting. It is commonly seen that teams start dividing their work very strictly. Resource gatherers, designers, builders, and communicators become separate roles with clearly defined interfaces and responsibilities. This proves effective in the context of the workshops but has negative effects on the truck factor (specialists missing at crucial points in the project) and could even be construed contrary to the notion of cross-functional teams. Thus, a workshop that encourages students to come up with such rigid structures might actually be contrary to the desired learning outcomes. We try to make this point when we take students out due to simulated "sickness" and discuss the consequences of this during the reviews. As discussed above, the teams' reactions are usually ad-hoc and no thoughtful redistribution of work is performed.

6. Theoretical Lens: Related Work

There are a few publications regarding the use of LEGO in higher education. One strand of research involves using LEGO Mindstorms NXT to introduce concepts in software development (e.g., Garcia and Patterson-McNeill (2002)), design concepts (e.g., Sharad (2007); Lew et al. (2010)) or engineering (e.g., Behrens et al. (2010)). While the authors emphasise the importance of team work, none of these papers uses Lego to introduce process knowledge with a low technical hurdle. In fact, Garcia and Patterson-McNeill (2002) advocate a waterfall process in their course. Lübke and Schneider (2005) describe a workshop to teach XP skills in 70 minutes. This is a refinement of the Extreme Hour (Merel, 2005) which tries to alleviate some of the problems of the original version that uses paper and pencil to build (images of) the final product. In the refined version, the participants build one single and rather small product (e.g., a family spaceship) from a set of requirements using LEGO bricks. The authors compare the performance of students and IT professionals in the exercise.

Another relevant strand of literature is concerned with teaching Scrum through serious games. Here we find contributions by Kropp and Meier (2014) and Kropp et al. (2014) who are inspired by Krivitsky's LEGO city game (Krivitsky, 2011) where Scrum is introduced by means of cooperating teams over multiple sprints. In Kropp's and Meier's version the LEGO is substituted for cardboard and other office material. von Wangenheim et al. (2013) describe a different setup where teams of students are asked to develop different paper artefacts over multiple sprints.

Among the many reasons for introducing Scrum using serious games, authors name that:

- games are more suitable for the restricted time constraints of a lecture (Lynch et al., 2011; Paasivaara et al., 2014; Kropp et al., 2014; von Wangenheim et al., 2013);

- the students get to apply Scrum in practice (Lynch et al., 2011; Paasivaara et al., 2014; Kropp et al., 2014; von Wangenheim et al., 2013); and

- being able to run multiple sprints within one session allows the students to evaluate their own process and the outcome (Lynch et al., 2011; Paasivaara et al., 2014).

In the intersection of the two strands we find two publications that use LEGO to introduce Scrum in an educational setting. These are the works of Lynch et al. (2011) and Paasivaara et al. (2014).

In Lynch et al. (2011), the authors describe their experience with teaching individual agile principles as well as parts of agile processes such as planning and review with Lego. A workshop format similar to the one presented here is used. They also report on intentional problems they have integrated — such as a disinterested customer — to create learning opportunities. An additional complication is the swapping of team members between iterations. The workshops are, however, less concerned with the collaboration between teams and solution-oriented thinking.

A game to teach Scrum using Lego is introduced in Paasivaara et al. (2014). The aim of the game differs slightly from ours and it is also set up differently. While we try to get students to understand the value of planning and customer collaboration by letting them fail and iterate with improvements, the game described in Paasivaara et al. (2014) is much more structured from the beginning. Each sprint follows a clearly defined schedule with a strong focus on planning activities. This also allows to include Scrum elements that we do not cover, such as the Daily Scrum and the creation of burn-down charts. Since each team is creating their own product, collaboration is only enforced within each team. The Scrum Master is not one of the students but one of the teachers. An important assumption is that students "have at least some

background knowledge in Software Engineering and basic knowledge on Scrum" (Paasivaara et al., 2014). Since we target beginners, we do not make such an assumption. The issues the authors see in their teams and the learning outcomes reported by the students are similar to those we observed. The paper makes no mention of how students applied Scrum in follow-up projects.

Finally, one example of papers that discuss teaching Scrum with a traditional project setting in higher education and how such a course can be supported by lectures is Werner et al. (2012). The students get to develop a project of their choice in groups of 5 to 7 and the development effort is prepared by introducing Scrum and a number of its practices, including planning poker and burn-down charts. In addition, videos, papers, and guest lectures are used to teach agile principles and the correct application of Scrum. The Product Owner and Scrum Master roles are filled by students within the project teams. The process is observed by the teachers through regular reports, including Sprint Plans and Sprint Reports as well as weekly reports about achievements. Over the course of the project, students report on problems and their solutions, but these seem to be focused on technical aspects rather than process aspects. Overall, both teachers and students report positively on the experience.

7. Discussion

The findings from the different lenses we have applied in the previous sections allow us to create a differentiated analysis of the current situation. We must acknowledge that our current practices do not fully allow us to reach our goals, but we also believe that there are indicators for a positive development and potential for further refinement of our methods and tools. The key point of the Lego Scrum workshops is that they allow us to be there when the students are applying the methodology and therefore give direct feedback on how it is applied. From this simple but powerful point arise a number of consequences that we will discuss in the following. We focus on the three research questions introduced in Section 1 and repeated here for clarity:

RQ1: What are the benefits and opportunities of using Lego workshops to teach Scrum?

RQ2: What are the trade-offs and limitations of using Lego workshops to teach Scrum?

RQ3: Which of the skills and knowledge from the workshop did the students transfer to their project work?

In the following, we will use the student lens (cf. Section 4), the autobiographical teacher lens (cf. Section 5), and the theoretical lens (cf. Section 6) to discuss our efforts and draw conclusions with regard to both benefits and opportunities as well as trade-offs and limitations. From the peer lens perspective, we also discuss the work that has been done in Augsburg, as it is sufficiently different to show complimentary results.

7.1. RQ1: Benefits and Opportunities

The analysis of the data shows that the exercise reveals a number of promising benefits in terms of teaching and learning Scrum. The overall assessment of the exercise from the peers that have had the opportunity to observe how we conduct the workshop is that they want to adopt it for their own educational purposes. In Delft the teachers are now conducting the workshop on their own while the teacher of the CE-T6 course is collaborating with one of the authors to make the workshop a regular event in the course. The following part of this section will now answer our first research question by detailing the benefits of the Lego workshop.

Changes in student confidence is a sign of reflection. As seen in Section 4.1.1 the students experience a positive change in confidence towards the skills needed to conduct Scrum in a systematic way. Coming back to Smith's definition of reflection, "what is in relation to what might or should be and includes feedback designed to reduce the gap" (Smith, 2001), the change in confidence implies that the students have reflected on their experiences during the workshop. The overall majority that report on a positive change in self-assessed confidence now have a better idea of what they are doing and *"what might or should be"* as well as an idea of how to get there.

Just as an increased confidence across the students is a positive effect of the exercise, the fact that students report a negative change in confidence shows that they have reflected on their experience. Here, the change reflects their understanding of the gap between "what is in relation to what might or should be" and that the feedback to bridge the gap is not necessarily that easy to implement. The data shows that many of the students that have a decrease in confidence are in the medium or high experience bracket. This corresponds to the learning curve where students realise that Scrum is more complicated than they thought. That the students with low experience of Scrum report fewer instances of lower confidence is not surprising since they have no or little initial idea of what the survey asks about.

Gradually increase complexity. The Lego exercise problematises important concepts such as understanding requirements, knowledge acquisition, communication with stakeholders, the definition of done, task estimation and breakdown, etc. By gradually increasing the number of concepts and the tools and skills for handling them the teacher can adjust the content of each sprint to balance the cognitive load towards the intended learning outcomes. The gradual increase is supported by the plan-act-reflect nature of the exercise and the Scrum methodology. We end up with two parallel cycles where the teachers' cycle

concerns the Lego exercise and the students' cycle considers Scrum and systematic sprint adaptation.

The evolving nature of the exercise also enables teachers to react to emerging learning opportunities as they happen. The events that trigger cognitive dissonance (Festinger, 1962) can either occur through the students new experiences (no more bricks, failing to deliver on time, lack of inter-team communication, ...) or by the teachers explicitly introducing them during the exercise (sickness, changing requirements, physical obstacles, ...). Either way, the exercise gives the opportunity to reflect-in-action (Schön, 1983) on the trigger and its consequences together with the students *when the dissonance occurs* instead of later in a different setting or from theoretical examples. In this way the exercise follows Kolb's learning cycle where experiential learning is combined with reflection and conceptualization (Kolb, 2014).

Emphasis goes from product to process. The agile manifesto emphasises the importance of delivering valuable software but also the importance of systematically developing software (Beck et al., 2001). Previous literature has reported on how teaching agile methods and practices tend to focus on the product increment and the tools and technologies used during the sprints (Alégroth et al., 2015; Steghöfer et al., 2016).

Drawing from the survey data reported in Table 2, the students' reflection reports, and our own observations we can see that the exercise clearly shifts the focus towards the Scrum methodology. The contrast is even more striking since the students report that previous experiences of applying Scrum have been in a context where the emphasis was on the product. According to Alégroth et al. (2015) and Steghöfer et al. (2016) the courses' aims explicitly included the learning of agile methods and practices. The Lego Scrum exercise is thus an answer to the open question of how to supply students with a complicated enough challenge while still focusing on a systematic approach instead of the product increment, assuring a *constructive alignment* between course aims and activities (Biggs, 1996). The positive shift towards process was also commented on by the teachers at the two extra-curricular events in the TI-T3 and CE-T6 cycles.

Plan for the future. The exercise supplies plenty of opportunities to emphasise concepts that are important to carry on from the Lego workshop to the software project. Both the retrospective and the review can be used to point out lessons for the future. The review is an opportunity to discuss the balance between quality and quantity and how testing is essential to ensure that the right increment is delivered with quality. This can be done by a simple test to see if a standard Lego figure fits at the steering wheel of the bus or only accepting buildings with uniform colour schemes. The review is also an opportunity to discuss the importance of delivering according to the prioritisation of the product backlog. The retrospective on the other hand can be used to discuss the importance of risk management such as dealing with sickness, time for knowledge acquisition in terms of understanding how to use new tools and technologies as well as how easy it is to forget new insights and misjudge simple tasks under stress.

The cyclic nature of the exercise also allows for trying out strategies to overcome perceived barriers before they are used in a project. In this way the exercise captures the reported benefits of learning Scrum through *serious games* (Lynch et al., 2011; Paasivaara et al., 2014) by giving a practical experience which can be reflected and acted upon in a series of sprints mirroring Kolb's *learning cycles* (Kolb, 2014) where the course project continues the series of cycles initiated during the Lego workshop.

Communication is a generic skill. The student reflections taken from the SEM-T4 exams show the importance of communication within the team, between teams as well as with the PO. Ensuring that students understand this point would not have been possible in a pure project setting where communication is very much within the team only. Communication is a generic engineering skill (Bennett et al., 1999; Young and Chapman, 2010) and there are multiple ways of introducing and teaching communication in an academic setting.

The benefit of the Lego Scrum exercise is that the (lack of) communication happens in front of and with the teacher. This means that communication is one of the topics that are covered by the increased learning opportunities and can be reflected upon together in a realistic context. Overall the self-assessed confidence rose for the students, but the negative change reported by 13 out of 100 students indicates that the exercise also shows the students that communication can be complex. Particularly the students with medium and high experience found communication challenging which shows that the training they have had until that point in time has not challenged their communication skills. Both the teachers at Delft, during the TI-T3 session, and the teacher at the CE-T6 session commented on how the workshop encourages the students to reflect on their communication skills. At Delft, the focus was on how the students could interact with other teams in a new way, encouraging playful interaction instead of competition, while the CE-T6 teacher saw the communication within the teams and the possibilities for reflecting on team work as the main contribution.

The workshop is unique in its kind. The students report that they have never had to communicate with so many parties in a project before, having to negotiate with both the PO and other teams. One of the issues that we had with teaching Scrum in a project context is that student teams do not interact in such settings normally, as each team works in isolation and is never forced to interact with other teams. There are no interfaces (software or organisational) between the work of the student teams and thus collaboration is not necessary. The Lego workshops force

the student teams to work together towards a common product composed of many individual parts. This fact surprises the students and shows them a new perspective on processes.

However, the students do not automatically perceive the uniqueness as an opportunity. First of all, they struggle to handle the cross-team collaboration, but second of all it also has little bearing on their future academic projects with a one-team-one-product setup. Thus the exercise reveals a fundamental shortcoming in their education and how it is misaligned with the Lego Scrum exercise and software development at large. The fact that neither Paasivaara et al. (2014) nor Lynch et al. (2011) have a setup requiring inter-team collaboration enforces our point that Software Engineering education is misaligned with industrial practices.

7.2. RQ2: Trade-offs and Limitations

While the data presented in this paper shows that Lego Scrum workshops are a promising way to teach the process without the need to address technical issues, the format has a number of limitations. In the following, these limitations will be discussed, drawing on the data previously presented and making recommendations on how they can possibly be addressed where appropriate. Many issues arise from the fact that it is impossible to cover all aspects in the workshop and—just like in a Scrum process—we need to decide how to spend the resources.

Workshops do not provide sufficient skills about task estimation and breakdown. Since the workshops do not emphasise task estimation and breakdown, the students do not understand the need for it and do not pick up the skills necessary to apply it in a project. This is a major difference of our workshop setup to the one used, e.g., by Paasivaara et al. (2014). However, the statements from the students, especially those from the IE-T6 reflection reports, clearly indicate that the students are not sufficiently equipped to plan and structure a sprint for a development project on their own. This is in contrast to the increase in confidence in the area that we can observe (cf. Table 1).

This issue must, however, be considered within the goals that are realistic for an initial Scrum workshop. For us, it is more important that students get to know a planning technique and understand the advantages and necessity of planning and breakdown than to become experts in these tasks.

Students do not feel sufficiently empowered to take responsibility for the process. Somewhat surprisingly for us as teachers, the students seem to have difficulties to accept their responsibility for the process. Many of the student comments in the SEM-T4 exams indicate that they were reluctant to take charge, e.g., to inspect the Lego before the workshop begins. Part of the workshops is to teach the students that they must do this to a certain degree, but the fact that they still do not shows that this aspect seems to be missing in the education the students have received up to that point. However, accepting responsibility is crucial when projects with limited supervision are carried out by the students.

Scrum as taught in the workshop is different from industrial practice. All of the points above allude to the biggest potential problem: the workshop experience is sufficiently different from the way Scrum is used in industrial practice. Ultimately, the goal of all Software Engineering education must be to impart students with the necessary skills and knowledge to allow them to work as productive members of a software development organisation. That means that students need to be prepared in a way that allows them to transfer their experiences from teaching to their future job. As argued above, the Lego Scrum workshops do not fully succeed in this endeavour. However, they acquaint students with the terminology, the principles, and practices that are used in similar forms in the industry. We therefore argue that even though there are substantial differences, the workshops still provide students with a framework that they can apply later on. In this light, however, there are two more issues that warrant discussion here: our use of user stories and the ambiguity between the teacher role and the Product Owner role.

One aspect of this is that some students also feel that the low-quality user stories used in the exercise make the problem harder than it should be. One student states that *"the backlog was used to force the issues to occur. A well written backlog with precise requirements would have helped prevent the technical debt that several groups carried through sprints two and three."* This is obviously a source of frustration and the student goes on to say that *"the creation of a deliberately poor backlog in order to force issues to occur was a strange decision."*

In addition, the way the PO is represented in the workshops does not conform to standard Scrum practice. Instead of being embedded in the teams and being available throughout the sprint, the PO must interact with all teams during the tightly packed sprint, introducing waiting times. In addition, the PO does not exhibit any kind of learning, e.g., with regard to the quality of the requirements (see the point above). On the contrary, the PO purposefully behaves in an almost "vicious" fashion to force learning opportunities to appear. The necessity of such a behaviour to achieve the learning outcomes has been acknowledged by others as well (see, e.g., Paasivaara et al. (2014)). Furthermore, since the PO and the teacher are the same person in our setup of the workshops, there might be confusion over the roles the teacher has in different situations. These roles should be clearly differentiated, e.g., by separating the feedback on the product during the Sprint Review from the feedback on the process.

For these reasons, there might be students that feel the workshop does not prepare them for using Scrum in subsequent projects. While the data indicates that this population is a minority, it must still be stated that the

format is not universally accepted as a valid teaching instrument by all students.

Uncertain long-term knowledge retention. Finally, it is uncertain how well the knowledge about Scrum is retained in the long-term. Our quantitative data was collected directly after the workshops and shows an average increase in *confidence* by the students, not in the level of skill and knowledge. Whether the knowledge acquired during the workshops is still retained after a year or longer, is however not clear from the data we have. What the data shows is that the knowledge is not actually applied correctly in the projects as discussed below. However, there are a number of confounding factors as discussed above that do not allow us to make a clear statement about knowledge retention just yet. We intend to evaluate the retention with students that went through the workshop in 2014 in a study in 2016, approximately 15 months after the initial workshop.

7.3. RQ3: Skills and Knowledge transfer to Project Work

Since one of the main purposes of our education in software processes is to allow the students to apply process skills and knowledge in actual development efforts, it is particularly interesting to see how well the transfer from the workshops into projects later on works. The results indicate that this transfer is fragmentary at best and that students have issues when asked to use Scrum within a larger scope. Our data also shows that student confidence is largely unaffected by project experience. We discuss these issues in the following.

Applying workshop knowledge to a development project is difficult. One of the findings that is evident in particular from the reflection reports by the IE-T6 students is that the knowledge acquired in Lego workshop can not be directly applied in a development project. The students report that the mapping from building a Lego city to actual development activities is not clear and that they often choose to change the process. It is not entirely clear what the reason for this problem is. One contributing factor could be that the workshops and the development projects have very different time spans and it is difficult to see how the stress of an 18 minute sprint translates into a sprint of two weeks. Another possibility is that the students seem to struggle most with planning and estimation tasks which are not the focus of the workshop. While there are students that acknowledge that estimations became better as the project progressed, many students quickly abandoned the practice due to negative experiences in the beginning.

Finally, the learning curve for the technical aspects students mention in their reports might overshadow the application of the process. In essence, students spend their time thinking about technical solutions and loose focus with respect to the tasks the process prescribes (Steghöfer et al., 2016). The survey data from the CEIT-T3 students supports this interpretation since they report a positive increase in confidence in using new technologies in future projects. In contrast they found it more difficult to interact with external stakeholders during their project. This might be explained by the fact that the teams are not co-located with all stakeholders and access to domain knowledge and customer perspectives is not readily available. Rather, it has to be planned and prioritised. This is mirrored in a recurring remark from the exercise retrospectives as the students reflect on that not talking to the product owner has a negative impact on product quality. And those IE-T6 teams that had a close relationship to the PO during the project reported at the final presentations that the sprint reviews had a positive impact on their understanding of customer value.

We have attempted to address the transfer of insights from the Lego exercise into the subsequent project with dedicated follow-up lectures in which we have made the translation into the project explicit and have given additional tips on how to do task breakdown and estimation. These efforts have not significantly improved the students performance, however, as witnessed by the statements from the IE-T6 students who received such a lecture. A different attempt is to ask students to write a plan of how to apply Scrum in the project to make them think about how they are going to use Scrum later on. No data is available yet to determine the outcome of this attempt.

Students omit crucial aspects of the process in the projects. The IE-T6 reflection reports make it particularly obvious that the students do not adopt Scrum completely. Instead, practically all the teams make adjustments to the process that can have detrimental effects. A particularly common adaptation is to abandon task breakdown and estimation (see discussion above). It must clearly be stated that we as teachers do the same thing, however. Due to the limited time for the sprints and the emphasis on practical aspects, some concepts and practices in Scrum are not or insufficiently addressed in the workshops. This includes the definition of done, burndown charts and task breakdown. Experiments with including these aspects in the workshop have proven unsuccessful, with students being confused, complaining about a lack of time, and missing the purpose of these practices. Burndown charts, e.g., are not particularly useful in the context of three or four sprints of 18 minutes each. It is difficult to apply task breakdown without explicit guidance and methodological background knowledge. We have therefore chosen to not address these topics in the workshops and rather cover them separately in lectures and exercises. However, their absence means that the workshops do not cover Scrum completely.

A major problem that can already be observed during the workshops is that students tend to meet the challenge by not doing better Scrum but by introducing strategies (like specialisation) that introduce single points of failure. While this is obvious in the teacher lens, it can also be seen in the SEM-T4 reports in which students were asked to report on how they improved the process in a second iteration. A common measure was to separate “builders”

and "gatherers". This specialisation is not, however, accompanied by communication and planning measures as they would be applied in a cross-functional team. This practice thus makes it possible to disable the team by taking individuals out of the running. In essence, the students make local, short-term optimisations and disregard the long-term effects. Similarly, the omission of practices such as task estimation can be seen as a short-term optimisation since the immediate effort seems too high and students do not see immediate benefits. Possible ways to address this are to interfere with such over-specialisations (even though we want students to take responsibility, see above) or to explicitly point out the problems this will cause in projects and emphasise the long-term benefits of using best practices.

7.4. Observations from Augsburg

In Augsburg, the Lego Scrum Workshop has been conducted for six years now. The workshop is offered as a part of a lecture on advanced Software Engineering topics in graduate programs in computer science. In contrast to the approach described above, in Augsburg Scrum is introduced in two lectures of 90 minutes each before the workshop is conducted. The lectures cover the process elements, the roles and the responsibilities of each role as well as practices commonly used by Scrum teams. The workshop is then used to reinforce the understanding of Scrum in the students. This slightly different approach leads to somewhat different observations. Since no surveys have been done with the participants in Augsburg so far, we can only report on the observations of the teachers. Most of the above-mentioned points, best-practices as well as learning opportunities, are also observed with two notable exceptions: students typically ask if they can see the Legos before starting the estimation, stating that the available bricks have a huge influence on what can be done; and students actively reorganize the room, remove obstacles and unpack the Lego the make their work easier.

These points show that the students understand at least some of the rights and responsibilities of self-organised teams. Nevertheless there are other points where the students fail in fulfilling their responsibilities as team member or Scrum Master. Most notably, the students typically accept far too large user stories instead of rejecting them and asking the Product Owner to cut them down.

Introducing large user stories into the product backlog is therefore somewhat ambivalent. On the one hand, they provide a learning opportunity: As team member I must make sure that too large user stories are rejected. But if the students fail in this, the large user stories make their way into the sprint backlog and create a problem there. The focus switches on getting the story done (product is in the focus) instead of correctly applying the process. This has the potential of ruining the first sprints. Such a rejection has never been observed in Gothenburg and the students are not made aware of the possibility to reject a user story there.

In Augsburg we are also trying to limit the number of concepts that the participants should learn. Due to the short sprints it was rather obvious that a stand-up meeting does not fit. Burndown charts were removed later. In the beginning we did use burndown charts, but the teams often ignored them as they did not see any value in them. One concept that we use in Augsburg that is not used in Gothenburg is the sprint goal. Our Product Owners do not simply give a list of backlog items which are rather arbitrarily prioritized. Instead we define a sprint goal for each sprint and prioritize the backlog items accordingly. Examples of sprint goals are "streets and mobility" and "housing and living environment". We did not use sprint goals from the beginning but added them to the simulation to make the Scrum simulation as realistic as possible by using as many standard Scrum practices as possible. Sprint goals are widely used by Scrum teams in industry but also suggested in, e.g., the Scaled Agile Framework (Leffingwell, 2013). Sprint goals help the POs to motivate their priorities for the stories and they have a catch phrase to communicate the expected result of the sprint.

We also learned after doing the first couple of instances of the workshop that having separate Product Owners and teachers is very important. Especially for giving feedback to the participants at the end of the workshop it is important for the teacher to look around and observe the participants while they work. That's very hard to do when the teacher is at the same time busy answering questions posed to him in his role as the PO. Furthermore having more people in the Product Owner role (we now have one teacher as PO per two student teams) streamlines the workshop. Phases with high PO-involvement (presentation of user stories and estimation) take a lot less time, compared to having only one PO for several student teams.

From our experience the points that were discussed in learning opportunities are more or less symptoms of the same fundamental problem: The participants do not fully understand what it means to work in a self-organising team and which responsibilities arise from committing to a sprint goal. The toughest role to fill is, of course, that of the Scrum Master. She has a lot of responsibilities which the participants usually have not experienced before. Being a good Scrum Master usually requires being an experienced developer and Scrum user. Both is typically not true for the workshop participants. In addition, the designated Scrum Masters have a hard time mapping their new responsibilities to a world of Lego bricks. Furthermore they tend to invest too much time building stuff and not enough time in supporting their team and watching for obstacles and interferences. Our introduction of the Scrum Master and his responsibilities in the preceding introduction to Scrum in lecture form so far did not do much to lessen the problems of the students in the Scrum Master role.

7.5. Validity Strategies and Threats

While our investigation combines both qualitative and quantitative data analysis our validity strategies have mainly

focused on the qualitative aspect. The qualitative analysis is a straightforward descriptive statistical analysis and we both describe when, how and why the data was collected as well reference to the on-line survey tool. This means the process behind the relevant data collection is transparent and possible to replicate. Still, the comparison of the confidence before and after the workshop is not a standardised knowledge test but rather a self-assessment by the students. As such, it does not show the actual knowledge levels, but rather how comfortable students feel with the various practices at that point in time. Since students made the post-workshop assessment right after the end of the workshop — which usually ends on a positive note with the successful delivery of the city — the self-assessment results might be influenced by the generally good mood the students are in when they filled out the questionnaire.

Regarding the qualitative analysis Guba defines four criteria for assessing the trustworthiness in qualitative analysis (Guba, 1981);

Credibility How well do the results mirror reality?

Dependability To which extent are the results dependent on the analytic process?

Transferability To which extent can the results be applied to a wider population?

Confirmability To which extent are the results dependent on the attitudes of the researchers?

We have addressed the dependability threat by detailing our methodology, as recommended by Shenton (2004), and by including the context of each action research cycle we strive to assist the reader to assess how the results will transfer to his or hers own educational situation (Lincoln and Guba, 1985). The following section will mainly focus on the credibility of the results, since credibility in practice has a strong influence on dependability (Lincoln and Guba, 1985)

Since two to three authors were present during the sessions of the first cycle, there were plenty of opportunities for reflection-in-action (Schön, 1983), i.e. to assess *"what is in relation to what might or should be and includes feedback designed to reduce the gap"* (Smith, 2001) while the exercise took place. Having more than one teacher present also meant that we could reflect from both a personal and a peer lens (Brookfield, 1995). It also gave us the possibility for reflection-on-action (Schön, 1983), where we shared and discussed our experiences after the sessions in order to improve how to carry out the exercises in the future. From a credibility aspect this gave us the possibility to familiarise ourselves with the educational benefits and drawbacks of the exercise (Lincoln and Guba, 1985). As we later moved on to carry out the exercise on our own we kept meeting the other authors to reflect on new observations, compare field notes as well as to analyse student behaviour and feedback. This mitigates the influence of individual researcher bias and has a positive impact on the confirmability of the results (Shenton, 2004).

As our understanding was solidified and we were more confident in how to implement and use the exercise, our data collection and analysis strategies became *deductive* in order to confirm and explain our data (Runeson et al., 2012). Finally, data from the last sessions were analysed to explore how the students transferred their gained skills and knowledge into the course project. Now we started to put our emerging themes in the light of the students' reflection reports, in particular to answer RQ3. Using reflection reports that are also used for grading as part of the qualitative analysis could introduce bias since students might write more positively about the workshops than they were actually perceived in an attempt to please the teacher who decides their grade. This threat to credibility was mitigated by focusing on the negative experiences and on reports that address shortcomings and limitations of the workshop. The analysis of the students' reports started out by using an *inductive* approach (or theory generating as Seaman (1999) puts it) to classify and structure the students' reflections which we then compared to our emerging themes. In this way we have followed the recommendations of Seaman (1999) to collect and analyse data in parallel in order to validate or refute emerging themes. Due to the action research nature of our approach, we have repeated both data collection and analysis multiple times, each time adjusting to the conclusions drawn from the previous iteration, testing themes until they mature or get dropped. Guba and Lincoln (1989) call the credibility strategy of continuously refining the analysis and best practices through reflection *member checking*.

The fourth author has not been part of the sessions that were held at GU and CTH. He has, however, held several sessions in Augsburg and contributed to this paper by comparing our experiences to his through a process of triangulation. He also helped set up the first workshop cycle in Gothenburg by offering his knowledge and insights about the workshops. By including other teachers in the exercise, as in cycle two and five, as well as externally auditing our findings in cycle five, we have used a credibility strategy recommended by Guba (1981) and Seaman (1999) to ensure that our method and findings resonate with our peers.

A credibility threat to the results is that we had to use convenience sampling instead of randomised sampling. Thus we have can have an uneven distribution of data sources so that certain aspects are more frequent – or not present – in our population than in general (Preece, 1994).

However, the nature of our action research process enabled us to collect our data at different locations, from a spread of people, and over time. The data is also taken from a multitude of sources including surveys, observations and student reports. The analysis of the data has been conducted independently by different authors who have then met and shared and compared the outcome. During the analysis four different lenses have been applied.

In this way we have *triangulation of data, investigators, methodology* as well as *theory* (Stake, 1995) to strengthen the credibility, transferability and confirmability of our results (Shenton, 2004). The output of our action research cycles is thus in line with Denscombe (2000), in terms of a sequence of changes to improve a given situation as well as generating guidelines for best practices.

8. Conclusion

In this paper, we have analysed our experience with teaching Scrum through workshops in which students build a Lego city. The quantitative data gives indications of student engagement and how their confidence in applying Scrum practices rises directly after the workshop. It also shows that confidence does not change significantly in a project that follows the exercise. Furthermore, our qualitative analysis sheds a light on how students apply the knowledge and skills they picked up in the workshops in an actual project setting. Our results show that the workshop format offers a number of benefits and opportunities, in particular by introducing students to the importance of communication in a project and by offering a number of learning opportunities that the teacher can react to directly. However, there are a number of limitations, in particular when it comes to transferring the workshops to a project. Most importantly, students dismiss crucial Scrum practices as irrelevant or too cumbersome.

For us as teachers, the workshops afford a new set of tools to interact with the students in multiple plan-act-reflect cycles. The shared experience with the students can be picked up later in the lectures and during the project to refer back to. This is in line with Kolb's learning cycle (Kolb, 2014). But we also see that the workshops are no silver brick: the limited duration and scope prevent the students from fully grasping all aspects of Scrum. Instead, they pick and mix specific practices and what they find convenient at the time, but miss the bigger picture of how the different Scrum practices complement each other. This means that no systematic way of working in the project emerges during the workshops or the following course project.

From the conclusions we draw, a number of open questions can be derived for future work:

How can we improve the structure of the workshops in order to improve a holistic adoption *of Scrum and especially mitigate the students' problems with task estimation and breakdown?.* We intend to go through additional cycles of our action research approach in order to find a solution for this problem. In particular, we are going to use the course instances in which the Lego Scrum workshop is given as an introduction to Scrum before a development project (like the IE-T6 course) to test different ideas. One first attempt will be to use specific exercises like the elephant carpaccio exercise described by Alistair Cockburn (Cockburn, 2010) and see if that has an effect on the task breakdown as reported by the students in the reflection report.

How does the workshop format influence long-term knowledge retention?. In order to determine how well the students retained knowledge after going through the Lego Scrum workshop, we intend to conduct a study in which we will let students from the first instance of the workshops in 2014 answer knowledge questions and present them with scenarios in which they should choose the correct action alternative based on their knowledge of Scrum. Since the students were asked to apply Scrum in at least two projects since they were first exposed to the process, this methodology is not suitable to explicitly test the effect of the workshops, but rather the knowledge retention the entire, problem-based curriculum provides. Nevertheless, this is a first step to see how the overall programme setup influences long-term knowledge retention of Scrum and agile principles. More focused follow-up studies will use the same technique with students that have not gone through a project, e.g., when the workshop was held before a semester break and the test is administered directly after the break.

How can we bridge the gap between the education the students receive in the workshops and the education they receive throughout their study programme to prepare them to become professionals?. The workshop is unique in the educational setting since students rarely apply inter-team communication in other courses. The workshops are conducted in rooms that resemble the physical working space at companies but that are not available to students when they pursue their projects. It also forces them to communicate with a Product Owner which is also unique in their studies. These characteristics are reasonable to prepare them for their jobs later on, but make the workshop poorly aligned with what is expected of the students in the rest of the study programme. We need to find ways to align the overall education on software processes more with what students will see in their professional experience and overcome the limitations set by the university resources and the current educational paradigm. We have started to address this in Steghöfer et al. (2016) and in the current iteration of the DAT255 Software Engineering Project. However, there is still a lot to be done to ensure that students not only leave the university with technical knowledge, but also with skills and knowledge for critical reflection and improvement, how to communicate with representatives from other disciplines, and how to systematically and sustainably deliver high quality in a team. This issue, of course, can not be resolved by tinkering with a workshop or a specific course, but requires a programmatic and paradigmatic shift.

Bibliography

Alégroth, E., Burden, H., Ericsson, M., Hammouda, I., Knauss, E., Steghöfer, J.-P., 2015. Teaching scrum – what we did, what we will do and what impedes us. In: Proc. 16th Int. Conference on Agile Software Development (XP2015). Vol. 212 of Lecture Notes in Business Information Processing. Springer.

Babb, J., Hoda, R., Nørbjerg, J., 2013. Barriers to Learning in Agile Software Development Projects. In: Baumeister, H., Weber, B. (Eds.), Agile Processes in Software Engineering and Extreme Programming. Vol. 149 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, pp. 1–15.

Bassey, M., 1981. Pedagogic research: On the relative merits of search for generalisation and study of single events. Oxford Review of Education 7 (1), 73–94.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D., 2001. Manifesto for agile software development. `http://www.agilemanifesto.org/`.

Behrens, A., Atorf, L., Schwann, R., Neumann, B., Schnitzler, R., Balle, J., Herold, T., Telle, A., Noll, T., Hameyer, K., Aach, T., May 2010. MATLAB Meets LEGO Mindstorms—A Freshman Introduction Course Into Practical Engineering. IEEE Transactions on Education 53 (2), 306–317.

Bennett, N., Dunne, E., Carré, C., January 1999. Patterns of core and generic skill provision in higher education. Higher Education 37 (1), 71–93.

Biggs, J., 1996. Enhancing teaching through constructive alignment. Higher Education 32 (3), 347–364.

Brookfield, S., 1995. Becoming a critically reflective teacher. Jossey-Bass, San Francisco.

Brooks, Jr., F. P., April 1987. No Silver Bullet—Essence and Accidents of Software Engineering. Computer 20 (4), 10–19.

Cockburn, A., 2010. Elephant Carpaccio Exercise. `http://alistair.cockburn.us/Elephant+Carpaccio+Exercise`, accessed 26th Nov 2014.

Cohen, L., L., M., Morrison, K., 2000. Research Methods in Education, 5th Edition. London: Routledge Falmer.

Denscombe, M., 2000. The Good Research Guide for Small Scale Social Research Projects. Taylor & Francis.

DeWalt, K., DeWalt, B., 2002. Participant Observation: A Guide for Fieldworkers. Anthropology / Ethnography. Rowman & Littlefield Pub Incorporated.

Dickens, L., Watkins, K., 1999. Action Research: Rethinking Lewin. Management Learning 2 (30), 127–140.

Dybå, T., Sjøberg, D. I. K., Cruzes, D. S., 2012. What works for whom, where, when, and why?: on the role of context in empirical software engineering. In: Runeson, P., Höst, M., Mendes, E., Andrews, A. A., Harrison, R. (Eds.), ESEM. ACM, pp. 19–28.

Elliot, J., 1991. Action research for educational change. McGraw-Hill Education (UK).

Festinger, L., 1962. A Theory of Cognitive Dissonance. Mass communication series. Stanford University Press.

Garcia, M., Patterson-McNeill, H., Nov 2002. Learn how to develop software using the toy lego mindstorms. In: Frontiers in Education. Vol. 3. pp. S4D–7–S4D–10 vol.3.

Guba, E., Lincoln, Y., 1989. Fourth Generation Evaluation. SAGE Publications.

Guba, E. G., 1981. Criteria for assessing the trustworthiness of naturalistic inquiries. Educational Technology Research and Development 29 (2), 75–91.

Kaplan, A., 1964. The Conduct of Inquiry: Methodology for Behavioral Science. Chandler Publishing Co., San Francisco, CA.

Kember, D., Gow, L., 1992. Action research as a form of staff development in higher education. Higher Education 23 (3), 297–310.

Kolb, D. A., 2014. Experiential learning: Experience as the source of learning and development. FT press.

Krivitsky, A., October 2011. Scrum simulation with lego bricks. URL `http://kiev2015.agileee.org/wp-content/uploads/2011/12/Scrum-Simulation-with-LEGO-Bricks-v2.0.pdf`

Kropp, M., Meier, A., April 2014. New sustainable teaching approaches in software engineering education. In: Global Engineering Education Conference (EDUCON). IEEE, pp. 1019–1022.

Kropp, M., Meier, A., Mateescu, M., Zahn, C., April 2014. Teaching and learning agile collaboration. In: IEEE 27th Conf. on Software Engineering Education and Training (CSEE&T). pp. 139–148.

Leffingwell, D., 2013. Scaled agile framework. http://scaledagileframework. com.

Lew, M., Horton, T., Sherriff, M., March 2010. Using LEGO MINDSTORMS NXT and LEJOS in an Advanced Software Engineering Course. In: IEEE 23rd Conference on Software Engineering Education and Training (CSEE&T). pp. 121–128.

Lewin, K., 1946. Action research and minority problems. Journal of social issues 2 (4), 34–46.

Lincoln, Y., Guba, E., 1985. Naturalistic Inquiry. Sage focus editions. SAGE Publications.

Lübke, D., Schneider, K., 2005. Agile Hour: Teaching XP Skills to Students and IT Professionals. In: Bomarius, F., Komi-Sirviö, S. (Eds.), Product Focused Software Process Improvement. Vol. 3547 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 517–529.

Lynch, T. D., Herold, M., Bolinger, J., Deshpande, S., Bihari, T., Ramanathan, J., Ramnath, R., 2011. An agile boot camp: Using a LEGO®-based active game to ground agile development principles. In: Frontiers in Education. IEEE, pp. F1H–1.

Merel, P., 2005. Extreme hour. `http://c2.com/xp/ExtremeHour.html`.

Moløkken-Østvold, K., Haugen, N. C., Benestad, H. C., 2008. Using planning poker for combining expert estimates in software projects. Journal of Systems and Software 81 (12), 2106–2117.

Paasivaara, M., Heikkilä, V., Lassenius, C., Toivola, T., 2014. Teaching students scrum using lego blocks. In: Companion Proceedings of the 36th International Conference on Software Engineering. ICSE Companion 2014. ACM, New York, NY, USA, pp. 382–391.

Preece, R., 1994. Starting Research: An Introduction to Academic Research and Dissertation Writing. Pinter Publishers.

Runeson, P., Höst, M., Rainer, A., Regnell, B., 2012. Case study research in software engineering: Guidelines and examples. John Wiley & Sons.

Schön, D. A., 1983. The Reflective Practitioner: How Professionals Think in Action. Harper torchbooks. Basic Books.

Schwaber, K., 1997. Scrum development process. In: Business Object Design and Implementation. Springer, pp. 117–134.

Seaman, C. B., Jul. 1999. Qualitative Methods in Empirical Studies of Software Engineering. IEEE Transactions of Software Engineering 25 (4), 557–572.

Sharad, S., June 2007. Introducing embedded design concepts to freshmen and sophomore engineering students with lego mindstorms nxt. In: Microelectronic Systems Education, 2007. MSE '07. IEEE International Conference on. pp. 119–120.

Shenton, A. K., 2004. Strategies for ensuring trustworthiness in qualitative research projects. Education for Information 22 (2), 63–75.

Smith, R. A., 2001. Formative evaluation and the scholarship of teaching and learning. New Directions for Teaching and Learning 2001 (88), 51–62.

Stake, R., 1995. The Art of Case Study Research. SAGE Publications.

Steghöfer, J.-P., Knauss, E., Alégroth, E., Hammouda, I., Burden, H., Ericsson, M., May 2016. Teaching Agile – Addressing the Conflict Between Project Delivery and Application of Agile. In: Software Engineering Education and Training Track, The 38th International Conference on Software Engineering.

von Wangenheim, C. G., Savi, R., Borgatto, A. F., 2013. Scrumia—an educational game for teaching scrum in computing courses. Journal of Systems and Software 86 (10), 2675 – 2687.

Werner, L., Arcamone, D., Ross, B., Apr. 2012. Using scrum in a quarter-length undergraduate software engineering course. J. Comput. Sci. Coll. 27 (4), 140–150.

Young, J., Chapman, E., 06 2010. Generic Competency Frameworks: A Brief Historical Overview. Education Research and Perspectives 37 (1), 1–24,124.