Interactive Learning – Increasing Student Participation through Shorter Exercise Cycles

Stephan Krusche
Technische Universität München
Munich, Germany
krusche@in.tum.de

Andreas Seitz
Technische Universität München
Munich, Germany
seitz@in.tum.de

Jürgen Börstler
Blekinge Institute of Technology
Karlskrona, Sweden
jurgen.borstler@bth.se

Bernd Bruegge
Technische Universität München
Munich, Germany
bruegge@in.tum.de

ABSTRACT

In large classes, there is typically a clear separation between content delivery in lectures on the one hand and content deepening in practical exercises on the other hand. This temporal and spatial separation has several disadvantages. In particular, it separates students' hearing about a new concept from being able to actually practicing and applying it, which may decrease knowledge retention.

To closely integrate lectures and practical exercises, we propose an approach which we call interactive learning: it is based on active, computer based and experiential learning, includes immediate feedback and learning from the reflection on experience. It decreases the time between content delivery and content deepening to a few minutes and allows for flexible and more efficient learning. Shorter exercise cycles allow students to apply and practice multiple concepts per teaching unit directly after they first heard about them.

We applied interactive learning in two large software engineering classes with 300 students each and evaluated its use qualitatively and quantitatively. The students' participation increases compared to traditional classes: until the end of the course, around 50 % of the students attend class and participate in exercises. Our evaluations show that students' learning experience and exam grades correlate with the increased participation. While educators need more time to prepare the class and the exercises, they need less time to review exercise submissions. The overall teaching effort for instructors and teaching assistants does not increase.

Keywords

Active Learning, Experiential Learning, Feedback, Reflection, Computing Education, Software Engineering

ACE '17, January 31-February 03, 2017, Geelong, VIC, Australia

ISBN 978-1-4503-4823-2/17/01…$15.00

DOI: http://dx.doi.org/10.1145/3013499.3013513

1. INTRODUCTION

When teaching large classes with hundreds of students, there is typically a significant delay between delivery of content in lectures and deepening and practicing of that content in follow-up exercises. The delay between lectures and exercises is usually a few days, up to a week (see Figure 1).

Figure 1: Delay between lectures and exercises in traditional learning in large class rooms

During this time, learners forget content that was discussed in the lecture. Participation, learning and knowledge retention might be reduced in both quantity and quality, when students are not cognitively active during content delivery and when there is a large time span between the content's delivery and actively dealing with it. This might lead to unnecessary knowledge gaps for the learners, especially if the interaction between educators and learners is low.

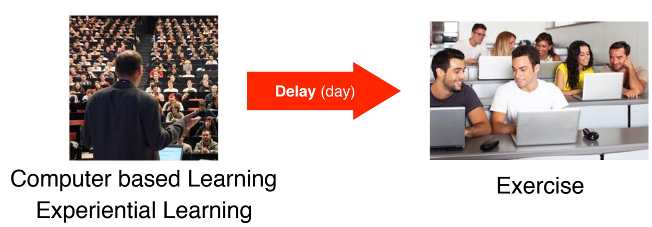

Figure 2: Reduced delay through the use of computer based and experiential learning

Computer based learning [8] and experiential learning [14] are approaches to reduce the delay between lectures and exercises as shown in Figure 2. Computer based learning supports students' learning in digital exercises and online media. Experiential learning creates opportunities to reflect on experience, a methodology in which educators engage with students to increase knowledge and develop skills.

Active learning is an educational approach to increase student involvement and excitement with the subject being taught by engaging students in activities [3]. **Interactive learning** is based on active learning, integrates computer based learning and experiential learning, immediate feedback and reflection. It tightens the relationship between content delivery and problem solving in class by integrating multiple, small units of content delivery and content deepening through exercises. By combining lectures and exercises into interactive classes, it reduces the delay to a few minutes (see Figure 3). Students reflect about the learned content immediately and increase their knowledge incrementally through a couple of short cycles covering theory, example, exercise, solution and reflection. Our hypothesis is that reducing the delay between lectures and exercises as proposed in interactive learning increases student participation in exercises and thereby improves the learning experience.

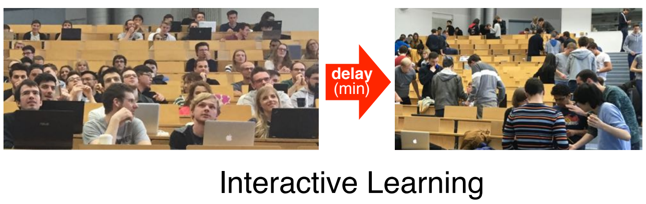

Figure 3: Interactive learning combines lectures and exercises into interactive classes and further reduces the delay to minutes

The paper is organized as follows: In Section 2, we describe the foundations in the areas of experiential and active learning. Section 3 presents the idea of interactive learning as an iterative process that combines lectures and exercises into short cycles. Section 4 shows a case study about two large software engineering courses, in which we applied interactive learning. In Section 5, we present the findings of qualitative and quantitative evaluations in these courses. Section 6 discusses related work and Section 7 summarizes the paper and presents the conclusion.

2. FOUNDATIONS

Exercises and examples are important elements in teaching and learning: "[E]xamples appear to play a central role in the early phases of cognitive skill acquisition" [27]. Just letting learners solve more problems is, however, not the most effective way to support learning. Carefully developed and integrated examples increase the learning outcome more than just letting learners solve more problems [25, 26]. In particular, in complex problem spaces, like software development, "[l]earners may learn more by solving problems with the guidance of some examples than solving more problems without the guidance of examples" [26].

Software engineering is an activity that requires collaboration and practical application of knowledge [29, 24]. Educators struggle when teaching it in traditional lecture based environments where activities take place in the front of the classroom. Lectures are usually similar to broadcasting, where essential education interactions take place initiated by the educator with only limited participation on the learners side. Self guided learning, personal responsibility, practical relevance and individualization are important elements of a great learning experience. Several pedagogic theories have been developed that include these elements.

Problem based learning is a technique to learn about a subject through problem solving. Educators facilitate by supporting, guiding, and monitoring this process [4]. While working in groups, learners identify what they know, what they need to know, and how and where to access new information that leads to the resolution of the problem.

Cooperative learning is an educational approach which aims to organize classroom activities into social learning experiences: learners work in groups to complete tasks collectively towards a common goal [11]. The educators role changes from giving information to facilitating learners' learning. Everyone succeeds when the group succeeds.

Experiential learning is the process of learning from experience, a methodology in which educators engage with learners in direct experience to increase knowledge, develop skills, and clarify values [14]. Aristoteles said: "For the things we have to learn before we can do them, we learn by doing them". John Dewey followed this idea with his statement that "there is an intimate and necessary relation between the process of actual experience and education".

Experiential learning is considered to be more efficient than passive learning like reading or listening. It is in contrast to academic learning where students acquire information through the study of a subject without the necessity for direct experience. The main dimensions of experiential learning are analysis, initiative, and immersion. Academic learning promotes the dimensions of constructive learning [28] and reproductive learning [12]. Both methods instill new knowledge, though academic learning makes use of more abstract techniques, whereas experiential learning actively involves the learner in a concrete experience such as an exercise.

Active learning is an educational approach to increase student involvement and excitement with the subject being taught [3]. Instead of students acting as receivers of knowledge by passively listening, active learning puts the emphasis on developing student skills and engaging them in activities such as small group discussions or a class game. Grabinger and Dunlop emphasize that authentic contexts encourage students to take more responsibility and engage students in learning activities that promote high level thinking processes [9]. An authentic context in software engineering would e.g. be the management of a project where students experience typical activities such as meeting and task management. Their learning progress is supported and assessed through realistic tasks such as planning and conducting a meeting or distributing tasks within the team.

Michael Prince examined the evidence for the effectiveness of active learning and discussed its common forms in [23]. He concluded that active learning positively influences knowledge transfer and student performance [23]. Joel Michael reviewed the literature and found that there is evidence that active learning improves the learning outcome compared to more passive approaches [22]. However, certain active learning approaches are not feasible for large class rooms. It is not possible to have a group discussion with 300 students in the same lecture hall. In addition, it is important that instructors "place a strong emphasis on guidance of the student learning process" to prevent misconceptions [13].

While the combination of these learning techniques leads to a more complex experience and to more effort for educators, it lowers their stress and leads to higher satisfaction [2]. A Chinese proverb, first mentioned by Confucius and adapted by Benjamin Franklin describes the underlying philosophy of experiential and active learning. In recent publications [15] an extended version of the proverb is mentioned: "Tell me and I will forget. Show me and I will remember. Involve me and I will understand. Step back and I will act."

"Tell me and I will forget" describes that explaining a concept only theoretically does not give learners the possibility to apply it. *"Show me and I will remember"* includes the idea of cognitive apprenticeship: an apprentice observes the skills of a master who shows how a concept works in practice, e.g. in a tutorial. Clarifying the thinking process behind the application of the concept makes it easier for the apprentice to imitate the behavior [7].

"Involve me and I will understand" includes aspects of active learning. Involving learners in the learning process allows them to apply a concept on their own, possibly in a different way that fits to their own techniques. It helps learners to understand a concept together with its application. *"Step back and I will act"* refers to self guided learning, self improvement and problem based learning. Learners take the responsibility to solve a certain problem on their own using the concepts they learned before. This proverb is the foundation of our teaching approach **interactive learning**.

3. INTERACTIVE LEARNING

Interactive learning aims to decrease the cycle time between teaching a concept and exercising it by combining lectures and exercises into interactive classes with multiple, short iterations. Thus, the typical separation between lectures and exercises disappears. We define interactive learning as follows: **Educators teach and exercise small chunks of content in short cycles and provide immediate feedback so that learners can reflect about the content and increase their knowledge incrementally.** Interactive learning expects active participation of learners and the use of computers (laptops, tables, smartphones) in classes. Instructors provide guidance during the learning process to facilitate learning and to prevent misconceptions.

Figure 4 shows the iterative process of interactive learning, where each iteration consists of five phases:

1. **Theory:** The educator introduces a new concept and describes the theory behind it. Learners listen and try to understand it.
2. **Example:** The educator provides an example, so learners can refer the theory to a concrete situation.
3. **Exercise:** The educator asks the learners to apply the concept in a short exercise. The learners submit their solution of the exercise.
4. **Solution:** The educator provides a sample solution and explains it to the learners. The educator can also show some exemplary solutions, submitted by learners and discuss their strengths and weaknesses to provide immediate feedback.
5. **Reflection:** The educator facilitates a discussion about the theory and the exercise so that the learners reflect about their first experience with the new concept.

In large education environments with hundreds of learners who participate in a course at the same time, teaching assistants (TAs) help in the conduction of the exercises. TAs walk through the classroom, answer questions and provide help in case problems occur or exercise instructions are unclear. The evaluation of the submitted solutions can either be automated using tool support or manually done by the TAs reviewing the submitted solutions and providing immediate feedback to the learners. The degree of automation depends on the exercise type and the format of the solution. The evaluation of programming assignments can e.g. be automated using continuous integration and test cases.

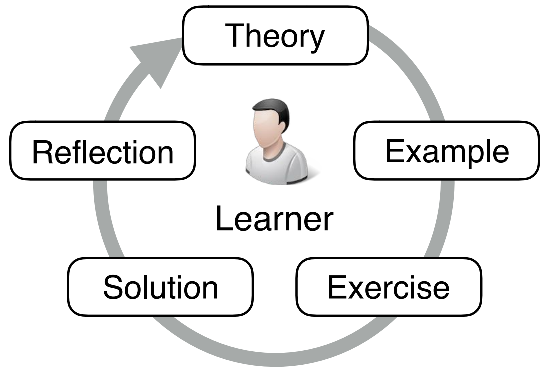

Figure 4: Interactive learning as iterative process

Interactive learning can be applied in **individual exercises** and in **team exercises**:

Individual exercises

- **E1** Quizzes with multiple choice questions (automatic evaluation through a quiz system)
- **E2** Interactive tutorials with step by step instructions (semi-automatic evaluation, degree depends on the exercise)
- **E3** Interactive coding challenges to solve programming assignments (automatic evaluation through test cases)
- **E4** Interactive modeling exercises (manual evaluation)

Interactive tutorials help students to directly experience a new concept. They are very detailed and include step-by-step instructions so that even beginners are able to conduct the exercise. The instructor conducts the tutorial live in-class so that students can follow the tutorial on their own computer. He asks the students several times during a tutorial how many can follow to synchronize the speed. TAs walk around and help students with problems. The instructor uploads slides with detailed screenshots before class, so that students can look up steps on the slides if they could not follow in the given time. More experienced students are kept motivated with optional challenges. Students who miss a class can catch up the exercise at home.

Team based exercises

- **E5** Project work that includes communication and collaboration aspects (semi-automatic evaluation, degree depends on the exercise)

Team based exercises also incorporate the concepts of peer learning and cooperative learning. They repeat topics of individual exercises to deepen and retain the knowledge by applying the learned concept in a different setting. Students have to transfer the knowledge they learned before to the concrete team situation and have to tailor the concepts. This facilitates self-guided learning and promotes the idea of self organization.

4. CASE STUDY

In 2015 and 2016, we applied interactive learning in two software engineering university courses "Software Engineering II: **P**roject **O**rganization and **M**anagement" (POM) and "**P**atterns in **S**oftware **E**ngineering" (PSE). In previous instances of these courses, content delivery (theory) and content deepening (exercises) were separated, i.e. students learned a concept in the lecture and applied it a week later in a central exercise session. In POM 2015 and in PSE 2015/16, we applied interactive learning and combined theory and exercises into interactive classes. Students learned a concept theoretically, then immediately applied it in a short exercise and received feedback about their progress.

4.1 Project Organiz. and Management (POM)

We taught POM in summer 2015 with 294 students. Typically, between 100 and 200 students attended class and participated in the exercises. They formed a heterogeneous group because the course was offered in multiple programs. Two distinct groups participated: (1) bachelor students in information science with a few experiences in software engineering and (2) master students in computer science with some existing experiences in the taught topics. This heterogeneity posed the challenge that students had a different velocity in completing class exercises. To improve this situation, the exercises included optional tasks specifically for more experienced students. In addition, the students had the opportunity to solve exercises as homework if they were not able to finish them in class. Some tasks and exercises were also explicitly designed as homework.

Week	Content
1	Team formation
2	Project organization
3	Software process models
4	Agile methods [18]
5	Prototyping
6	Proposal management
7	Branch & merge management [19]
8	Contracting
9	Continuous integration
10	Continuous delivery [17]
11	Risk and demo management
12	Global project management [21]
13	Project management antipatterns

Table 1: Overview of the course content in POM

The module description of POM describes the following intended learning outcomes. Participants understand the key concepts of software project management. They are able to deal with problems such as writing a software project management plan, initiating and managing a software project and tailoring a software lifecycle. They are familiar with risk management, scheduling, planning, quality management, build management and release management, and can apply these techniques to solve simple problem. Table 1 shows the schedule and the content of the lecture.

While the exercise participation was optional for the students, they could earn bonus points for completing exercises successfully in order to improve their grade in the final exam. This motivated the students to participate in the individual and team based exercises. In POM, we used the exercise types quizzes (E1), interactive tutorials (E2) and project work (E5) as described in Section 3.

Interactive Tutorial

Students had to solve individual tasks on their computer. They cooperated with the instructor, TAs and fellow students to solve particular problems. They learned from their experience in exercises and reflected about the concepts they just learned before. The instructor conducted four large interactive tutorial in POM using dedicated tools:

1. Agile Methods (Atlassian JIRA[1])
2. Branch and Merge management (Atlassian Bitbucket[2])
3. Continuous Integration (Atlassian Bamboo[3])
4. Continuous Delivery (HockeyApp[4])

In these interactive tutorials, the instructor introduced concepts and immediately applied them in short exercises. The students completed the exercises on their own computer using the mentioned tools in the browser. During each tutorial, the students either looked at the detailed slides that were handed out at the beginning of the exercise or watched how the instructor conducted the exercise on the presentation computer. In addition, TAs walked through the lecture hall, helping students by answering questions directly.

Each interactive tutorial consisted of three to five exercises which were decomposed into smaller tasks. In summary, the students had to solve about twelve to twenty tasks in one tutorial. The instructor synchronized the speed of the tutorial several times by asking students about their progress and by checking the number of participants and results in the tools. If more than about 90 % were able to complete a particular tasks, the instructor proceeded to the next exercise.

As an example, we describe the execution of the two exercises about continuous integration and continuous delivery based on the release management workflow described in [17]. The instructor mapped an exemplary delivery process for a mobile application to the continuous integration server that was used in class, Bamboo. To simplify the exercises, each student first forked a preconfigured source code repository and cloned a preconfigured build plan. Then, the students adapted and configured the build plan, fixed existing test cases and wrote additional test cases. A change in the requirements of the software led to a bug that was detected by Bamboo during a regression test and fixed by the students so that all tests passed again at the end of the exercise and the students could deliver the software to their fellow students who played the role of test users.

Project Work

In addition to the individual exercises, the students participated in a team project (exercise type E5) with five team

[1]http://www.atlassian.com/software/jira
[2]http://www.atlassian.com/software/bitbucket
[3]http://www.atlassian.com/software/bamboo
[4]http://hockeyapp.net

members, a simplified version of the team projects described in [5]. The goal of the project was that the students experience the learned concepts in a more realistic environment. The instructor played the role of the customer and provided three short problem statements about the development of mobile applications. The teams had to choose one of the problem statements and a development environment and target platform, either Android, Windows Phone or iOS.

The instructor arranged the students into different teams according to their self assessment. The goal was to have balanced teams with respect to the skill level of the students. Team based exercises also built on experiential learning techniques. However, they had a stronger focus on problem based and cooperative learning. Software engineering is a collaborative activity [29], therefore team work is an important skill students have to learn. The teams used Rugby [18] as agile and continuous process model with an initial warm-up phase and five development sprints.

In addition, students only received a vague description of the exercises that deliberately missed detailed instructions so that the teams have to think on their own about how to solve the exercise. This approach follows the principle "Step back and I will act" of the Chinese Proverb in Section 2.

Students first learned and experienced concepts in individual exercises. Then, they applied the knowledge in team exercises to improve their long term memory. They had to tailor the concepts to their concrete team situation and had to agree upon different decisions in their team which facilitates communication, collaboration and conflict handling. In later classes, students reflected about their team experiences.

4.2 Patterns (PSE)

We taught PSE in winter 2015/16 with 324 students. Typically, between 150 and 250 students attended class and participated in the exercises. The course included key concepts of different types of patterns that can be used during software development, in particular design patterns, architectural patterns, testing patterns, antipatterns, and organizational patterns. The learning goals are that students understand patterns as a way to describe reusable knowledge for analysis, system design, object design and software project management activities. Given a problem, they are able to identify the applicability of a pattern that addresses the problem, describe the pattern in UML and map it to Java source code. The course was attended by bachelor and master students mainly of the field of computer science. Table 2 shows the schedule and the content of the course.

Week	Content
1	Introduction and pattern definition
2	Basic concepts
3, 4, 5	Design patterns
6, 7	Architectural patterns
8, 9	Antipatterns
10, 11	Testing patterns
12	Pattern based reengineering
13	Global software engineering

Table 2: Overview of the course content in PSE

During the 13 classes, we conducted 39 exercises in total, 29 of them were in-class exercises, 10 of them were homework. In contrast to POM, we did not conduct any team exercises (E5) in PSE. We focused on the exercises types interactive coding challenge (E3) and interactive modeling challenges (E4). We also carried out quizzes (E1) and interactive tutorials (E2). In the homework exercises, students further deepened their knowledge. In PSE, we did not offer bonus points, so students could not improve their final exam grade. Exercise participation was optional for the students.

Interactive Coding Challenge

The participants had to write new source code or adjust existing one, commit their changes to a version control systems which then automatically triggered test cases on a continuous integration server to verify the given solution. To increase the extrinsic motivation, the first three students submitting a correct solution have been rewarded by the lecturer in form of gummy bears or donuts. In addition, there was a wildcard winner which has been randomly picked and acknowledged without being among the first three correct submissions.

We applied continuous integration with unit tests to verify the submitted solutions of the students automatically and immediately. Coding exercise were distributed with a version control system. We used Atlassian Bitbucket as repository and Atlassian Bamboo as integration server. The needed material for the specific exercise was provided beforehand the lectures in a repository accessible to all students. To synchronize the working time on the exercise, access to the material was secured by a password.

After introducing the theory, explaining the problem and providing the corresponding password, the students started to work on the exercise. With this approach, we made sure that all students started working on the exercise at the same time and had the same timeframe for solving the problem. The timeframe to submit the exercise was determined by the instructors and the elapsed time during the exercise was visualized on a big stop clock on one of the projectors. Once the deadline had passed, the instructors provided a sample solution and discussed it with the students. Winning students also had the possibility to explain their solution and why they came up with this approach. Due to the fact that we used a version control system and continuous integration, it was possible to track the participation and validate the results of the students in real time.

As an example we describe the conduction of an exercise regarding the state design pattern that we conducted in lecture 4. After the introduction and explanation of the state design pattern, the exercise for the students was to implement a basic remote control for a TV with four states. The exercise was to apply the state pattern to implement the transition between the appropriate states. The instructors provided a standard Java Eclipse project with existing source code, unrelated to the state pattern itself. The problem was visualized in a UML state diagram describing the different state transitions and its limitations.

The instructor set the timeframe to solve this exercise to 15 minutes, released the password and the students started to work on the exercise. During the work on the exercise, the TAs helped the students in case there were any questions regarding the exercise. Students could ask for help by raising their hand. After 5 minutes, a hint was given to the students in form of a class diagram representing the implementation of the state pattern. In this exercise, 145 students took part using 145 repositories and 145 automatically configured build plans. Each submission led to an execution of 20 test cases, resulting in 2900 test results within 15 minutes. As the deadline for working on the exercise had lapsed, the first three correct submissions and the wildcard winner were honored. The instructors discussed a sample solution with the students to reflect about the concept of the state pattern.

5. EVALUATION

In both courses, we evaluated the effects of interactive learning by investigating whether there is a correlation between exercise participation and the students' grade in the final exam, which represents the competence and gained knowledge of students in the taught topic. In addition, we conducted an online questionnaire in POM, where we asked students about their personal opinion on their improvements in a specific technique and their confidence to apply the technique in a later situation.

5.1 Study Design

We state the following three hypotheses with respect to interactive learning:

- **H1 Participation**: Interactive learning increases the participation of students in classes.
- **H2 Improved Learning**: Interactive learning leads to an improved learning experience for students.
- **H3 Scalability**: Interactive learning is scalable to large classes with 300 students.

We validated the hypotheses with a qualitative evaluation in POM and quantitative evaluations in both courses. The qualitative evaluation was conducted as online questionnaire. We investigated the students' improvements and confidence in the techniques that we applied in the individual and team based exercises. After the end of POM, we invited 294 students, who wrote the final exam of the course, to participate in the questionnaire. The anonymous questionnaire consisted of six closed questions, took about five minutes and was not mandatory for the students.

The first two questions were about personal data, field of study and degree. The third question asked whether students participated in specific individual exercises, the fourth question asked whether they applied specific techniques in their team project. The last two questions used a five point Likert type scale with the answers *strongly disagree*, *disagree*, *neutral*, *agree*, *strongly agree* to measure either negative, neutral or positive responses. The fifth question measured if students were able to improve their skills in these techniques and the sixth question measured if students are confident to apply these six techniques in their next team project.

We conducted the survey in July 2015 and gave students two weeks to complete it. We created personalized tokens and send them to the exam participants of the course. The open source survey tool LimeSurvey[5] guarantees that the answers are still anonymous by strictly separating token and answer tables in the database. We sent two reminders to the students who did not yet participate. We combined the responses of the results into a three point Likert type scale with positive responses (*strongly agree* and *agree*), neutral and negative ones (*strongly disagree* and *disagree*) to minimize positive and negative outliers.

We conducted two quantitative evaluations to investigate the learning experience in both courses. The quantitative measurement focused on the relationship between exercise participation and the final grade of the students. We calculated the exercise points in POM and the exercise participation in PSE for each participating student and correlated it with the final exam grade. We grouped the students after the relative exercise points / participation into five categories, calculated the grade point average for each category and computed the correlation using a χ^2 test.

[5] http://www.limesurvey.org

5.2 Findings

We first present the qualitative findings of the online questionnaire. 223 out of 294 students (76 %) participated in the questionnaire at the end of POM. As described in more detail in Section 4, the students first learned concepts in individual exercises conducted in class. Then, they applied these concepts in team exercises. The questionnaire asked the students whether they participated in four individual exercises and whether they applied these four techniques in their team: agile methods, branch and merge management, continuous integration and continuous delivery.

The questionnaire included the following two statements for each of these techniques:

1. **Improved**: In the exercises, I was able to improve my skills in the technique
2. **Confident**: I am confident to apply the technique in my next team project

In the following, we describe the answers to these statements for students who participated in individual exercises **and** who applied the technique in their team.

Figure 5 shows that 85 % of the students perceived that they improved their skills in agile methods and that 88 % of the students are confident to apply agile methods in their next team project. This result confirms the strong focus of the exercises on agile methods and shows that students feel prepared for the management of their next agile project.

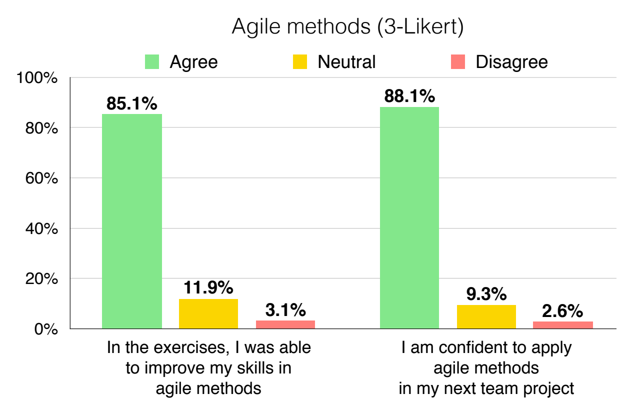

Figure 5: Perceived improvements and confidence in *agile methods* in POM

Figure 6 shows that 78 % of the students perceived that they improved their skills in branch and merge management. 87 % of the students are confident to apply it in their next team project. These results show that branch and merge management have become teachable. Students are able to handle multiple branches and can deal with merge conflicts.

Figure 7 shows that 76 % of the students perceived that they improved their skills in continuous integration. The

Content	(1) Individual exercise			(2) Team based exercise			(3) Both exercise types		
	#	Agree improved	Agree confident	#	Agree improved	Agree confident	#	Agree improved	Agree confident
Agile methods	209	83.7 %	87.1 %	198	84.8 %	88.4 %	194	85.1 %	88.1 %
Branch & merge management	197	75.6 %	81.7 %	162	75.9 %	85.8 %	155	78.1 %	86.5 %
Continuous integration	166	71.7 %	73.5 %	138	70.3 %	73.9 %	119	75.6 %	76.5 %
Continuous delivery	149	71.8 %	71.8 %	118	73.7 %	73.7 %	98	77.6 %	78.6 %

Table 3: Percentage of students who participated in exercises and perceived that they *improved* their skills respectively perceived that they are *confident* to apply the technique in their next project

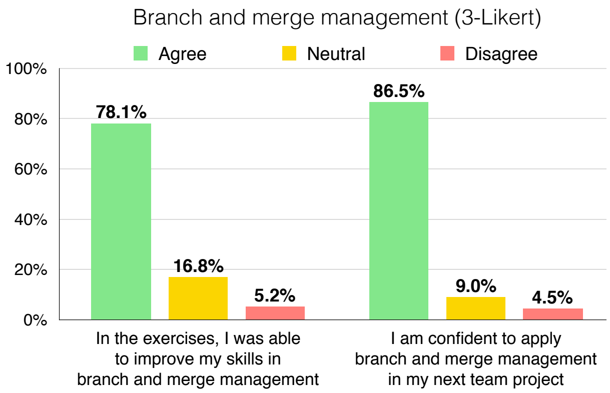

Figure 6: Perceived improvements and confidence in *branch and merge management* in POM

students could experience the benefits of immediate feedback about integration and test failures after they committed their changes to the source code repository. 77 % feel confident to apply continuous integration in their next team project.

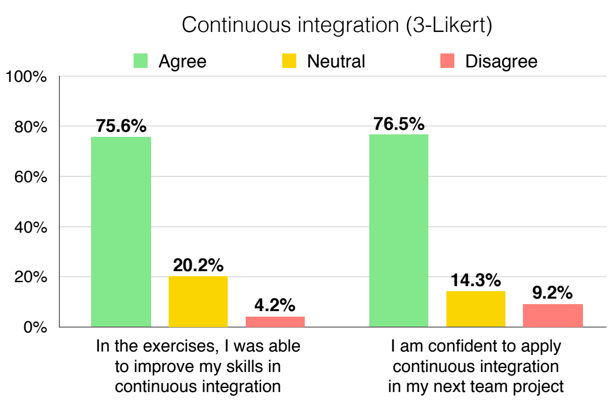

Figure 7: Perceived improvements and confidence in *continuous integration* in POM

Figure 8 shows that 78 % of the students perceived that they improved their skills in continuous delivery. In the corresponding exercise, they configured continuous delivery for a mobile application and applied it in their team project as well. 79 % of the students are confident to apply continuous delivery in their next team project.

In Table 3, we summarize the evaluation results of the questionnaires for the four techniques. The table shows that the participation in individual exercises was higher than in

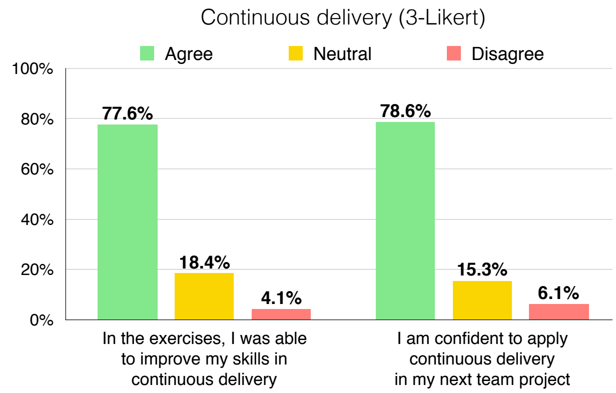

Figure 8: Perceived improvements and confidence in *continuous delivery* in POM

team based exercises. While 209 students (71 %) participated in the individual exercise on agile methods, 198 students (67 %) participated in the team exercise, and 194 students (66 %) in both exercises.

In one of the last exercises about continuous delivery, there were still 149 students (51 %) participating in the individual exercise, 118 students (40 %) in the team exercises and 98 students (33 %) in both exercises. The numbers were lower, because the exercise was more challenging in complexity and required more effort by the students. Table 3 includes the following three different filters:

(1) **Individual exercise:** We considered students who reported that they participated in the individual exercise of the corresponding technique.
(2) **Team based exercise:** We considered students who reported that they applied the concept in their team project.
(3) **Both exercise types:** We considered students who reported that they participated in the individual exercises **and** who applied the technique in their team. These are the same results as shown in Figure 5 - Figure 8.

In addition to the qualitative evaluation, we also looked at attendance rates. Figure 9 shows that the number of participants per lecture was relatively stable throughout the POM course in 2015. The number of students decreased from around 60 % in the beginning to around 50 % in the end of the course, although classes started at 8:15 am in the morning and the in-class quizzes were conducted in the beginning of the class[6].

[6]Some students missed these quizzes as they came late to the 8:15 am classes, so the actual attendance rate was higher.

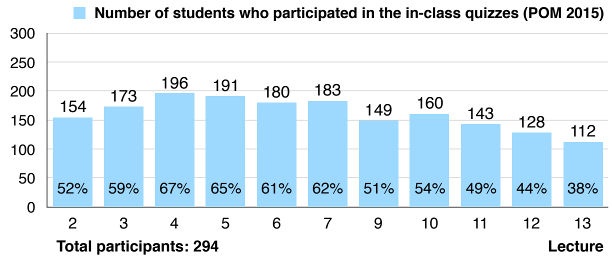

Figure 9: Number of participants per lecture in POM in summer term 2015

The attendance rate in 2015 was higher than in the previous instance of the course offered in 2014 (compare Figure 10), when the number of students who visited a lecture dropped to below 20 %[7]. The attendance rates in 2014 are more in line with other courses at our faculty that are taught in a more traditional way. This indicates that interactive learning might help to increase the participation of students in classes (H1). The increase in participation might, however, be a result of other factors. Further investigations are therefore needed.

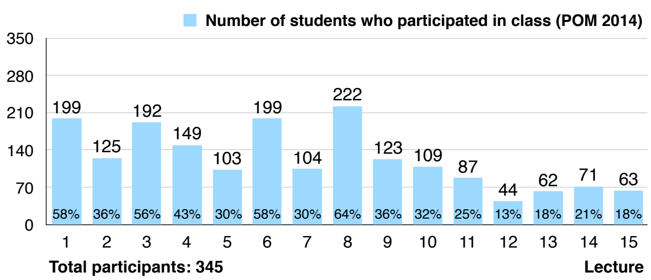

Figure 10: Number of participants per lecture in POM in summer term 2014

We evaluated whether there is a correlation between exercise participation and the average grade of the students in the final exam. The students could receive up to 600 bonus points through the participation in quizzes, individual exercises and team based exercises.

We grouped the 294 students, who participated in the exam, into five categories with equal distances describing their participation in the exercises, see Table 4 and Table 5. For instance, the first category in POM contains 75 students who obtained less than 20 % of the exercise points and the second category contains 66 students who obtained between 20 % and 40 % of the exercise points. Figure 11 show that students with lower exercise participation have worse grades than students with a higher participation.

In fact, students who successfully participated in less than 20 % of the exercises have a grade point average (GPA) of 3.9 in POM and a GPA of 3.1 in PSE. Students with more than 60 % have a GPA of 2.5 or better in POM[8] and a GPA of 2.1 or better in PSE.

[7]Some lectures in 2014, e.g. 1, 3, 6 and 8, were between 12:15 and 13:45 and had higher attendance rates than the other ones starting at 8:15 am.

[8]The GPA in Figure 11 and Table 4 does not include bonus that students could obtain through exercise participation.

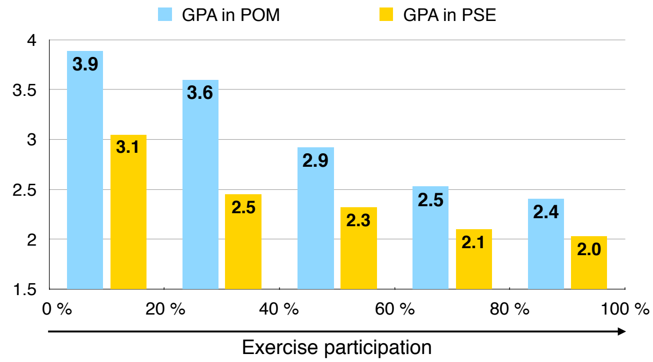

Figure 11: GPA (grade point average) of the final exam of POM (blue) and PSE (yellow) grouped by students' exercise participation. Final exam grades vary between 1.0 and 5.0 and do not include any bonuses; lower grades are better.

A χ^2 test [1] shows that there is a strong and highly significant correlation between exercise participation and the grade in the final exam in POM ($\chi^2 = 82.53; p < 0.0001$) as well as in PSE ($\chi^2 = 48.01; p < 0.0001$), see Table 4 and Table 5. Students who participate more in the exercises tend to achieve better grades in their final exams. This might indicate a positive effect of interactive learning (H2).

However, it could also mean that stronger students (with good grades) tend to participate more in the exercises compared to weaker students (with lower grades). We have no data from previous courses to evaluate the effects of other factors such as motivation or previous experience of students on exercise participation and exam results. Further studies are therefore needed. From the number of students in our two case studies, we also conclude that instructors can apply interactive learning (as implemented in our setting) in large courses with 300 students, which supports H3.

5.3 Limitations

In our qualitative evaluation in POM in summer 2015, one threat to the validity is that the personal opinion of students might not reflect the real situation, because it is subjective. Most students were beginners in the taught concepts and reported about their perceived improvements. A student without previous knowledge in an area will improve his knowledge, even if he only learns a limited amount of concepts. Beginners might not be able to objectively estimate their improvements in a subject. The confidence to apply a concept does not necessarily mean that the student is really able to apply it.

Other variables of the course, such as an open atmosphere towards feedback, have a positive influence on the evaluation result. If a student likes interactive exercises, it does not necessarily mean that he improves his skills. We were not able to exclude these variables in the evaluations of the case study. To alleviate these threats, we additionally evaluated the participation in the exercises and the correlation of students exercise participation and exam results quantitatively.

In the quantitative measurements, we recognize the following threats to validity. The participation in interactive exercises might be one reason for the found correlation that leads to the improvement of the final exam grade, but the

Exercise points (relative)	0 - 20 %	20 - 40 %	40 - 60 %	60 - 80 %	80 - 100 %	**All**
Number of students with very good grade (1.0 - 1.3)	0	0	8	10	2	20
Number of students with good grade (1.7 - 2.3)	5	11	22	21	3	62
Number of students with satisfactory grade (2.7 - 3.3)	23	21	35	12	2	93
Number of students with sufficient grade (3.7 - 4.0)	14	16	14	12	2	58
Number of students who failed (4.3 - 5.0)	33	18	9	1	0	61
Number of students who wrote the exam (Sum)	75	66	88	56	9	294
Grade point average (GPA) without bonus (all students)	3.9	3.6	2.9	2.5	2.4	3.2

Table 4: Correlation between exercise participation and GPA in the final exam in POM: students who successfully completed more exercises received more exercise points and scored significantly better in the exam (grades vary between 1.0 and 5.0; a lower grade is better, a higher is worse).

Exercise participation (relative)	0 - 20 %	20 - 40 %	40 - 60 %	60 - 80 %	80 - 100 %	**All**
Number of students with very good grade (1.0 - 1.3)	9	13	12	9	4	47
Number of students with good grade (1.7 - 2.3)	45	34	15	18	7	119
Number of students with satisfactory grade (2.7 - 3.3)	44	25	8	8	5	90
Number of students with sufficient grade (3.7 - 4.0)	20	6	6	3	0	35
Number of students who failed (4.3 - 5.0)	28	3	2	0	0	33
Number of students who wrote the exam (Sum)	146	81	43	38	16	324
Grade point average (GPA, all students)	3.1	2.5	2.3	2.1	2.0	2.7

Table 5: Correlation between exercise participation and GPA in the final exam in PSE: students who participated in exercises scored significantly better in the exam (grades vary between 1.0 and 5.0; a lower grade is better, a higher is worse).

correlation does not necessarily show this causality. Other causes of the correlation might be the motivation or the previous knowledge of the students. Students who participate in exercises more frequently usually also have a higher motivation to learn the theory for the exam, or they might have more previous experience. We were not able to measure these variables in the presented courses or to exclude them and therefore more detailed studies are necessary to investigate the actual effects of them on the learning outcomes.

6. RELATED WORKD

There are several courses in computer science that apply active learning techniques. They report an increase in students' learning, engagement, and performance. We look at three approaches and compare them with our approach.

Kurtz et al. describe an active learning approach using microlabs [20]. Students perform short activities during a lecture, either individually or in groups, and submit their answers to an automated grading system. They receive constructive feedback, and can revise their answers. Kurtz et al. conclude that microlabs can increase the students learning experience. Their approach can be compared with in-class exercises in interactive learning. In our case, students also have a time limit for exercises and submit their solutions to an automated grading system, or upload them for manual assessment if no automated assessment is possible. Both approaches have in common, that they are used in class during lectures. Heckman reports that there is "a large increase in student engagement" for the use of in-class labs [10].

Another popular approach is think pair share (TPS), where students work on a problem individually, then in small groups and finally reflect about it with the whole class. We propose a similar approach where students first experience a concept individually and then apply it within a team. Kothiyal et al. describe a large programming course which uses TPS [16]. The course includes programming labs and lectures with two TPS activities: students worked on questions individually first, and then in pairs, while the instructor helps in case of questions. Class wide discussions were facilitated concerning the former tasks. The study reports an average of 83 % student engagement for TPS based activities. This approach shows parallels to our course setup, since we introduced individual and team exercises, similar to the think and pair phases. Interactive learning also covers the share phase, as students could join a discussion with the instructor.

Campbell et al. describe an approach based on the flipped classroom concept with video lectures, labs and assignments [6]. They also used quizzes, contributing to the course grade as we propose for our approach. However, the authors do not use in-class exercises and report a low lecture attendance rate. Our approach implements frequent homework assignments, as well as immediate feedback for exercises to keep students motivated. Campbell et. al. suggest to use such an approach instead of labs for future improvement.

7. CONCLUSION

In this paper we described interactive learning, an approach based on active learning, computer based learning and experiential learning for large class rooms with guidance and immediate feedback. In interactive learning, the educator delivers small chunks of content and exercises incrementally, in short cycles, so that learners reflect about the content immediately. This can increase students' participation, can support knowledge deepening and can improve knowledge retention. By reducing the delay between theory, example, exercise, solution and reflection to a few minutes, we establish a tighter integration of lectures and exercises leading to interactive classes.

We applied and evaluated interactive learning in two large software engineering classes with 300 students each. In the quantitative evaluations, we found a strong and highly significant correlation between exercise participation and the final exam grade. Students who participate more in the exercises also achieve better grades in their final exams. A qualitative evaluation shows that students perceive that they improve their skills and feel able to apply these skills in later projects. This indicates that interactive learning might improve students' learning experience and learning outcomes. More studies are, however, necessary to investigate the role of interactive learning in those results.

Our case studies show that interactive learning is scalable and applicable to large classes without increasing the teaching effort significantly. We will introduce interactive learning into more courses. We believe it can help to bring an interactive component to massive open online courses (MOOCs) where we want to mix videos and fully automated interactive exercises to improve the students' learning experience and the adaptivity of the course with respect to heterogeneous student groups.

8. REFERENCES

[1] M. Abramowitz, I. A. Stegun, et al. Handbook of mathematical functions. *Applied mathematics series*, 55:62, 1966.

[2] R. Ben-Ari, R. Krole, and D. Har-Even. Differential effects of simple frontal versus complex teaching strategy on teachers' stress, burnout, and satisfaction. *International Journal of Stress Management*, 2003.

[3] C. Bonwell and J. Eison. *Active Learning: Creating Excitement in the Classroom*. ASHE-ERIC Higher Education Reports., 1991.

[4] D. Boud and G. Feletti. *The challenge of problem-based learning*. Psychology Press, 1998.

[5] B. Bruegge, S. Krusche, and L. Alperowitz. Software engineering project courses with industrial clients. *ACM Transactions on Computing Education*, 2015.

[6] J. Campbell, D. Horton, M. Craig, and P. Gries. Evaluating an inverted cs1. In *Proceedings of the 45th technical symposium on Computer science education*, pages 307–312. ACM, 2014.

[7] A. Collins, J. S. Brown, and A. Holum. Cognitive apprenticeship: Making thinking visible. *American educator*, 1991.

[8] R. Garrison and H. Kanuka. Blended learning: Uncovering its transformative potential in higher education. *The internet and higher education*, 2004.

[9] S. Grabinger and J. Dunlap. Rich environments for active learning: A definition. *Research in learning Technology*, 3(2), 1995.

[10] S. Heckman. An empirical study of in-class laboratories on student learning of linear data structures. In *Proceedings of the 11th annual International Conference on International Computing Education Research*, pages 217–225. ACM, 2015.

[11] D. Johnson, K. Smith, and R. Johnson. Cooperative learning: increasing college faculty instructional productivity. *ASHE-ERIC higher education reports.*, 1991.

[12] S. Jong, A. Jan, R. Wierstra, and J. Hermanussen. An exploration of the relationship between academic and experiential learning approaches in vocational education. *British Journal of Educational Psychology*, 76(1):155–169, 2006.

[13] P. Kirschner, J. Sweller, and R. Clark. Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational psychologist*, 41(2):75–86, 2006.

[14] D. Kolb. *Experiential learning: Experience as the source of learning and development*, volume 1. Prentice Hall, 1984.

[15] J. Korthagen, Fredand Kessels, B. Koster, B. Lagerwerf, and T. Wubbels. *Linking practice and theory: The pedagogy of realistic teacher education*. Routledge, 2001.

[16] A. Kothiyal, R. Majumdar, S. Murthy, and S. Iyer. Effect of think-pair-share in a large cs1 class: 83% sustained engagement. In *Proceedings of the 9th annual international conference on International computing education research*, pages 137–144. ACM, 2013.

[17] S. Krusche and L. Alperowitz. Introduction of Continuous Delivery in Multi-Customer Project Courses. In *Proceedings of the 36th International Conference on Software Engineering*, pages 335–343. IEEE, 2014.

[18] S. Krusche, L. Alperowitz, B. Bruegge, and M. Wagner. Rugby: An agile process model based on continuous delivery. In *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*, pages 42–50. ACM, 2014.

[19] S. Krusche, M. Berisha, and B. Bruegge. Teaching Code Review Management using Branch Based Workflows. In *Companion Proceedings of the 38th International Conference on Software Engineering*. IEEE, 2016.

[20] B. Kurtz, J. Fenwick, R. Tashakkori, A. Esmail, and S. Tate. Active learning during lecture using tablets. In *Proceedings of the 45th technical symposium on computer science education*, pages 121–126. ACM, 2014.

[21] Y. Li, S. Krusche, C. Lescher, and B. Bruegge. Teaching global software engineering by simulating a global project in the classroom. In *Proceedings of the 47th SIGCSE*, pages 187–192. ACM, 2016.

[22] J. Michael. Where's the evidence that active learning works? *Advances in Physiology Education*, 30(4):159–167, 2006.

[23] M. Prince. Does active learning work? a review of the research. *Journal of Engineering Education*, 93(4):223–231, 2004.

[24] D. Shaffer. Pedagogical praxis: The professions as models for postindustrial education. *Teachers College Record*, 106(7):1401–1421, 2004.

[25] J. Sweller and G. A. Cooper. The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, 2(1):59–89, 1985.

[26] J. G. Trafton and B. J. Reiser. Studying examples and solving problems: Contributions to skill acquisition. Technical report, Naval HCI Research Lab, Washington, DC, USA, 1993.

[27] K. VanLehn. Cognitive skill acquisition. *Annual Review of Psychology*, 47:513–539, 1996.

[28] J. D. Vermunt. The regulation of constructive learning processes. *British journal of educational psychology*, 68(2):149–171, 1998.

[29] J. Whitehead. Collaboration in software engineering: A roadmap. *FOSE*, 7(2007):214–225, 2007.