Industrial Case Studies in Graduate Requirements Engineering Courses: The Impact on Student Motivation

Marian Daun, Andrea Salmon, Bastian Tenbergen, Thorsten Weyer

paluno – The Ruhr Institute for Software Technology
University of Duisburg-Essen, Germany
{marian.daun | andrea.salmon | bastian.tenbergen | thorsten.weyer}@paluno.uni-due.de

Abstract

University education in software engineering instructs sound theoretical concepts together with method competence. It seeks to provide hands-on experience with the learning content along with insights into its application in practice. Even theoretical disciplines are beginning to adopt more experience-oriented instruction as opposed to passive, lecture-oriented instruction. One favored way for experience-oriented instructions is using case studies in lecture-accompanying assignments and/or tutorials. Compared with real-world scenarios, such case studies are often simplified in order to illustrate specific challenges related to the instructed material. This paper reports on our experience in using realistic industry-oriented case studies in a requirements engineering course with graduate students. The experience indicates a strong positive effect on student motivation as well as the degree of comprehension of the instructed theoretical material. These findings are confirmed by evaluations of the learning experience as self-reported through students’ questionnaires. Comparing the exam results with previous years indicates substantial improvement in final exam scores.

1. Introduction

University education in the software engineering discipline suffers from a serious tension between teaching theoretical, sound concepts and training industry-related practical skills. This gap between academia and industry is currently, if at all, only insufficiently addressed (cf. [1, 2]): While current software engineering education at large places emphasis on theoretically sound curricula, it does not seem to cater towards industry needs, particularly with regard to practical aspects (cf. [3, 4]). Thereby, students neither experience the full complexity related to a project nor the evolving and partly escalating nature of software engineering projects [5]. Furthermore, passive instruction in general may lead to a lack of social and personal skills necessary to work in teams, which nowadays is a core facility graduates must possess for real-world development projects [6]. Another facet of the gap between academia and industry, as reported in [1, 2], is that graduates may have been instructed on theoretical concepts and are familiar with academic examples but are lacking an in-depth understanding of industry challenges and require additional instruction until their academic knowledge can be applied in industrial setting [6]. University level software engineering education is currently undergoing a change. Many educators are moving away from the conventional classroom teaching and apply non-traditional teaching methods to make education more effective and interesting for the students [6-12].

In this paper, we present our experience from applying more interactive and industry-oriented teaching techniques to a graduate requirements engineering course. The combination of current teaching techniques is aimed at (i) improving students’ understanding of industry and their problems, (ii) enabling students to deal with real
problems by applying academic methods and concepts on their own, and (iii) to motivate students. To do so, we developed industry-oriented case studies in close collaboration with industrial partners. In the following, Section 2 briefly summarizes the relevant state of the art. Section 3 outlines our teaching approach and explains the nature of the case studies. In Section 4, the instructional setting in which our teaching approach was applied is described. Section 5 reports on the results and our experiences gained. Section 6 discusses the results and thereby concludes the paper.

2. Related Work

In university software engineering education, the traditional and most common approach is conventional lecturing: students attend to readings or presentations passively, are asked to complete additional reading assignments, and take exams at the end of the course [3, 12]. Albeit frequently applied, this passive learning approach has several disadvantages: students are not actively involved in the learning process [12, 13] and hence settle for short-term objectives (i.e., passing the exam with a satisfying grade) instead of leveraging on long-term learning goals [3, 12], leading to poor long-term retention of the instructed material [3, 13]. To overcome students’ passiveness, problem-based learning is often used [12]. Student teams are asked to solve problems with regard to some problem domain. They do so by identifying gaps in their knowledge necessary to solve the problem and acquiring this information on their own. However, the problem domains are often very small in scope to fit into an academic context, which in some cases limits the number of possible solutions and may impair students’ industry-readiness after graduation (cf. [3]). Furthermore, students often lack enthusiasm for the instructed material and, in particular, find theoretical problems in theory-heavy subjects boring and unexciting (e.g. [7], [8]).

One way to address this issue is seen in project-based learning (e.g. [9], [10]): students are asked to develop, implement, or create a case example in a self-directed manner by making use of material that has been instructed in different classes or throughout their degree programs or complete milestones and can hence focus on problem solving rather than rote memorization [14]. While project-based learning is seen as sparking interest and enthusiasm in students, it lacks in conveying difficult theoretical relationships [15].

Another paradigm to foster industry-oriented university learning are case example-oriented approaches (see e.g. [11] and [12]). In these instructional settings, students also work in small teams. In contrast to project-based instruction, students are not asked to develop a concrete project themselves but are given excerpts from real life projects conducted in industry and are asked to complete assignments based on the real-life case example. Exemplary student solutions are typically discussed in class, which is meant to facilitate learning and communication skills. Most case example-based approaches aim at guiding the students to the intended solution, as opposed to project-based learning approaches where solutions are meant to be found freely by the students themselves.

In addition to these general concepts, research on requirements engineering education has addressed making the educational experience more realistic by incorporating real stakeholders (e.g. [16]). The focus therein was on realistic requirements elicitation and documentation. The impact on student motivation and on fostering students’ awareness for realistic industry challenges was not reported on.
## 3. Teaching Approach

As we have illustrated in Section 2, there is a number of teaching approaches and paradigms that aim at bridging the gap between theory-driven academic university instruction and a more practice-oriented, industry-driven instruction. Since each of the above-mentioned approaches has both advantages as well as disadvantages, we decided to use a combination of well-known techniques and realistic examples from our industrial partners in a Masters’ level requirements engineering course to achieve three main goals:

- **Industry-Orientation.** The students shall gain awareness of industrially relevant problems. The students shall not only acquire sensitivity for industrial development scenarios, but become familiar with realistic industrial challenges, tasks, and problems in requirements engineering.

- **Method Competence.** While industry-orientation is often applied at the expense of theoretical method competence, we explicitly wanted to foster an in-depth understanding of requirements engineering theory. Students shall be motivated through industry-orientation, gain knowledge of theoretical concepts and methods, and develop enthusiasm for the instructed material.

- **Problem-Solving Skills.** The students shall not only be aware of industry problems and have knowledge about theoretical concepts and methods. Students shall also be able to use the academic concepts to solve real-world problems on their own. Therefore, it was also necessary to foster team-work and communication skills.

To reach those goals, we chose a combination of problem-based, project-based, and case example-oriented learning techniques. For this purpose, we developed case examples and case studies in close collaboration with industry partners and adapted them for the course. We extensively discussed current as well as general problems in industrial scenarios with our industrial partners and ensured the appropriateness of the case examples by reviews and revisions by our industrial partners. In the following, we briefly describe the individual techniques that were combined in the Masters’ level requirements engineering course:

- **Lecture.** Since lectures are still the common approach to university education and are seen as successful in soundly teaching theoretical concepts. The course thus consisted of a series of classical lectures. To improve the student involvement we enriched the lectures with frequent questions directed to the student audience. In addition, we illustrated the theoretical concepts by adding examples from our industry cooperation. We constantly encouraged students to interject any kind of question at any point.

- **Tutorials & Assignments.** We designed biweekly assignment sheets discussed in dedicated, weekly tutorial sessions. To support the students’ understanding the assignments focused on problems of low to moderate difficulty which the students could solve using the instructed material from the lecture. In the tutorial sessions, we encouraged students to discuss the assignments, but we never collected nor graded them.

- **Case Examples.** Case examples were developed in an iterative process in cooperation with industry partners. The objective of the case examples was to transfer research results from a federally sponsored project into industrial practice. The case examples were hence especially suited to highlight specific industry challenges and were therefore selected for the RE course. Industry partners were available throughout the course to answer specific
questions about the case examples and industry challenges. Each case example dealt with one single automotive embedded system (e.g. an adaptive cruise control).

- **Case Studies.** In the case studies, students were asked to complete consisted of the case examples discussed above and a series of milestones that obligatorily had to be submitted for review and critique. Students were introduced to the case examples during the tutorial sessions and some introductory material was provided. Students self-selected into teams of four to six and chose one of the available case examples that they would work on throughout the whole semester. The milestones were the same for each case example to ensure that each case study had a similar level of difficulty. The milestones represented major development phases or techniques and comprised concepts covered during several lectures. To help students find their own solutions for problems, they were allowed to revise or resubmit milestones as needed.

- **Presentations & Discussions.** To give insights in the multiplicity of problems and possible solutions, each team had to present their milestone accomplishments to the other teams during the tutorial sessions. Furthermore, the teams were encouraged to present preliminary work as well, as we intended to foster discussion among the students and to allow students to engage themselves in problems and concepts.

- **No Sample Solutions.** We did not provide sample solutions in order to encourage students to come up with individual solutions on their own. Instead, as discussed above, assignments sheets and project milestones were discussed in class where solutions to individual problems were developed in the tutorial sessions by the entire student body, supported by the tutorial instructor. In cases the students came up with more than one correct solution, the tutorial instructor directed the discussion to advantages and disadvantages of the solutions to illustrate which solution fits which situation best.

### 4. Course Design

In Section 5, we discuss our experiences and some empirical data we gathered from the application of the teaching approach described in Section 3. In order to do so, we give background information on the course itself in this section. Section 4.1 gives information about the participating student population. Section 4.2 discusses the lecture and topics discussed therein. Since the changes with regard to the teaching approach mostly pertain to how the tutorial session was organized, Section 4.3 illustrates how the tutorial was conducted before the teaching approach was introduced, and Section 4.4 outlines how the tutorial was organized according to the new teaching approach.

#### 4.1. Students

From 2010 to 2012, about 25 students took the course each year. The teaching approach was first introduced in 2012, after which the number of participating students increased to about 50 students in 2013. Students are typically first and second year Masters’ level students enrolled in degree programs for either software engineering or business information systems. Undergraduate students may be admitted to the course if provisionally accepted into either Masters’ program. In both degree programs, the course is offered for elective credit, i.e., students may participate in the course in partial fulfillment of their respective degree requirements. Due to privacy restrictions, demographic data may neither be recorded nor reported here. Attendance in neither the lectures nor the tutorials was compulsory.
4.2. Lecture

The course aims at teaching advanced methods and techniques pertaining to the documentation, and analysis of textual and model-based requirements. The course builds upon foundations instructed in a companion undergraduate course that teaches the basic concepts of requirements engineering. The graduate course comprises 15 weeks, each week offering one lecture and one tutorial session, each of which lasts 90 minutes. The following material is covered: goal- and scenario modeling, essential systems analysis, requirements validation, and requirements management (including prioritization and negotiation). The grade is determined by a final exam at the end of the semester. However, in order to be admitted for the exam, successful completion of the tutorial session is mandatory.

4.3. Tutorial until 2011

Prior to the case study-based paradigm, the course tutorial consisted of three blocks, totaling 150 points. In the first block, students were asked to prepare presentations on papers discussing topics that were covered in class up to that point. The aim was that every student makes herself familiar with a particular topic such that she can present it to the other students. Students were scored with a maximum of 30 points for the paper presentation block. In the second block, students were instructed in the usage of commercial tools (i.e. IBM Doors as well as SparxSystems Enterprise Architect) that can be used gainfully in requirements engineering and asked to complete a couple of toy examples. The tool block was also scored with 30 points and employed a short multiple choice quiz covering questions concerning how the respective tools can be used to achieve certain tasks pertaining to the topics covered in the lecture. The aim of this block was to foster hands-on experience. In part three, students were given a total of six mandatory assignment sheets over the course of approximately six to eight weeks. Students were asked to prepare solutions for the assignments within one week in teams of two. Assignment sheets were graded with 15 points per sheet, totaling 90 points for this block. In the week after assignment sheets were due, graded solutions were returned to the students and the correct solutions were discussed during the tutorial sessions. Throughout the entire semester, students were given the opportunity to meet with the tutorial instructor upon request to discuss open questions about lecture material or the grading of the assignment sheets. For admittance to the exam, students had to achieve a minimum of 60% of points throughout the tutorial.

4.4. Tutorial in 2012 and 2013

Since the summer term of 2012, the three-block paradigm of the tutorial has been abandoned in favor of the teaching approach discussed in Section 3: Instead of being offered three blocks of material, students were asked to self-select into respective teams that were given the chance to pick one of the case examples, as explained above. It was ensured that every team worked on a different case study to enforce variety in the solutions and in discussions. Teams were asked to complete a total of four incremental milestones. The milestones consisted of tasks that had to be fulfilled for the respective project and were the same across all case studies (e.g., “For your [system name], develop and document the relevant scenarios that refine the goals from the previous milestone”). Each milestone was submitted electronically and reviewed by the tutorial instructor. Students were asked to rework incorrect solutions and to resubmit for additional review. By design, case study solutions were not predetermined since the aim
was to let the students explore possible options to complete specific tasks pertaining to lecture material (see Section 3). Hence, there was no sample solution. The solutions presented by the students were neither considered right nor wrong (beside obvious contradictions, model inconsistencies, syntactical errors, or the like), but can be regarded as well or poorly handled. Milestone solutions were discussed in class, where case study teams presented solutions to one another. Furthermore, case study teams were given the opportunity to discuss their preliminary work on the milestones with the whole course, including the tutorial instructor. Throughout the semester, six assignment sheets were given out biweekly. To contrast with the embedded system case study, the assignment sheets solely focused on information systems. However, in contrast to the prior modus operandi (see Section 4.3), the assignment sheets were not submitted for grading. All assignment sheets were discussed in the tutorial, similar to prior to 2012 (see Section 4.3). For admittance to the final exam, participation was considered successful if all case study milestones were passed.

5. Experiences and Results

This section discusses experiences and results we gained after applying the case studies in the Masters’ level requirements engineering course. In the following section, we will outline qualitative observations about student motivation and learning success. Afterwards, we present quantitative data pertaining to how students evaluated the teaching approach based on a questionnaire study in Section 5.2 and report on the evolution of students’ exam results using the new teaching approach in Section 5.3.

5.1. Some Qualitative Experience

As we have discussed in Section 3, we placed strong emphasis on self-directed learning and discussions in the new teaching approach to foster both enthusiasm as well as student motivation. After implementing the new teaching approach in 2012, we noticed a clear trend in the nature of the discussions: while we encouraged discussions prior to 2012 as well (i.e., with regard to assignment sheets and their solution), discussions after implementing the new teaching approach in 2012 and 2013 tended to be more thematically involved. Before, discussions dealt mostly with minor, mostly technical issues (e.g., exact notations, or the interpretation of the wording of assignments). After implementing the case study-oriented approach, the nature of discussion shifted away from technical issues towards more theory-prone topics (e.g., the correct interpretation of modeling language semantics or different solution alternatives to convey the same meaning). In particular, using the new approach, discussions frequently pointed out possible misinterpretations by specific stakeholders. Hence, solutions to minimize ambiguity were developed in the tutorials. Furthermore, discussions frequently evolved beyond the scope of the case studies. While in 2011 and before discussions were limited to the solution of the assignment sheets, in 2012 and after discussions often contrasted the characteristics of embedded systems and information systems and often picked up how “things would be in a real-life development project” as a central theme. This was surprising because other than informing the students at the beginning of the course that the case studies are real-world industry examples, we made no further effort to emphasize real-world applications throughout the course.

In general, we observed that student participation during discussions increased by a great deal: While before only 5 or 7 students out of 25 would regularly and actively
involve themselves in discussions, almost the entire class would actively participate using the new approach. In addition, we expected that only few students would work on the voluntary assignment sheets, however, almost all students asked questions about assignment sheets at least once during the tutorial sessions. Furthermore, as outlined in Section 4.4, students were given the opportunity to present their preliminary milestones to the entire class to get feedback from other students as well as from the instructor. This opportunity was extensively used by the students, partly resulting in extended lessons of up to 150 minutes instead of the scheduled 90 minutes. Discussions in class aimed at understanding regarding the academic problem as well as the industrial challenge, the discussion of possible solutions and the question of which solution is appropriate under specific circumstances. As a result, all students passed all milestones in the end. Only in one case, a student team had to revise a milestone three times. Yet, this did not result in student irritation.

The aim for this approach was to help students gain a better understanding of industrial applicability, problems, and method limits. We observed in student solutions that rather than simply completing the project milestones, a variety of assumptions were documented which allowed the students to complete the milestones in their own way. These assumptions typically clarified the students’ understanding of the milestones and were meant to allow the students to cope with individual challenges arising from the nature of their case study. For example, when asked to model an engine control unit, one student team made the assumption that a CAN-bus like communication system exists in the remainder of the car. These assumptions typically allowed students to focus on more challenging aspects of the case study rather than spending much time on issues that are considered mundane.

Lastly, we observed highly increased participant numbers in the course in 2013 after first applying the new concept in 2012. The rise from about 25 students to about 50 students (see Section 4.4) can possibly be explained by the application of the new teaching approach: As we will see in Section 5.2, the teaching approach was rather popular among students, which may have increased course popularity among prospective students, resulting in higher enrollment rates the following year.

### 5.2. Findings of the Annual Student Evaluation

At the end of each term, the students are given the opportunity to evaluate their courses and give critical feedback regarding the instruction they received. These course evaluations are conducted on the authority of the university’s teaching quality assurance program and are administered by the course instructors. The evaluation consists of about 20 questions answered on a 5-point Likert-scale (“5” representing complete disagreement, “1” representing complete agreement) and features five open-ended questions. In this section, we present quantitative data on some evaluation questions.

Please note that the evaluation questionnaire changes slightly from year to year. Therefore, we chose the subset of questions that were repeatedly asked every year. In addition, questions were disregarded if they could not be related to changes in the teaching concept (e.g., questions regarding spatial arrangements or equipment provided in the classroom, or questions dealing with the students’ personal circumstances).

The diagram in Figure 1 depicts the evaluations of seven comparable questions we chose from the evaluation sheets, from 2010 to 2013. As can be seen, the students evaluated the course and its impact in 2012 on average better than in previous years. In particular, we noticed that the students’ self-reported learning experience across the board is considerably higher than it used to be. Considering that the instructed material
did not change between the years, we feel confident that this outcome can be attributed to the modifications in the instructional approach. However, Figure 1 also shows a slight decrease in the students’ positive evaluation of the course in 2013. Yet, the course was still evaluated better compared to 2010 and 2011. This may be attributed to the increased enrollment numbers in 2013. We were unprepared for the increased enrollment numbers and were unable to offer additional tutorial groups with additional instructors. This led to a decreased instructor-to-student-ratio. In consequence, we will take this into account for 2014 and thereby validate whether this assumption is correct.

**Figure 1: Students’ self-reported learning experience from 2010 to 2013**

In addition to the learning experience, the evaluation questionnaire features a number of questions with regard to methodological instruction as compared to other department courses. In the department where the graduate requirements engineering course is taught, more than 5,000 students are enrolled in over 200 courses for 15 degree programs (the University of Duisburg-Essen is one of Germany’s largest universities with over 39,000 enrolled students). As Figure 2 depicts, the graduate requirements engineering course was evaluated far better than the arithmetic average of the remaining departments’ courses.

**Figure 2: Comparison of the RE course and other department courses in 2012**

In particular, the results in Figure 2 show that self-guided learning and the ability to grasp theoretical concepts showed an improvement over the remaining department courses. Once again, since the course is neither harder nor easier to pass than other department courses and since the education within the department is typically considered to be of high quality, we assume that the application of the new teaching approach is responsible for this outcome.
5.3. Evolution of Students Exam Results

We furthermore analyzed the written exams from 2008 to 2013 in order to see if the teaching approach had an impact on the overall success in the course. The diagram in Figure 3 shows the minimum, the first quartile, the arithmetic average, the third quartile, and the maximum of the achieved exam grade in percent each year. In this diagram, an effort was made to normalize the data. For example, students who failed the exam with “zero” percent were filtered out, as for traditional reasons, the degree programs allow students “free failures”, i.e. to intentionally fail an exam without penalty. Hence, it is common for students to participate in the exam in order to get accustomed to the types of questions asked and then fail the exam with zero percent of attainable points by simply handing in unanswered exam sheets. Students can then take a make-up exam for full credit a few weeks later.

![Figure 3: Development of Student Exam Results (in percent of maximum attainable points)](image)

The diagram shows that from 2008 on, the percentage of achieved exam scores increases gradually. In 2008 and 2009, the graduate requirements engineering course was still a required course for all degree programs, however, it was changed to an elective course in 2010. This might explain the comparatively poor exam results in 2008 and 2009, as also students with no interest in requirements engineering participated in the course. Results from 2012 show that after adopting the new teaching approach students performed as well as they did the year before. This means that the new teaching approach did not lower the students’ ability to learn the theoretical concepts. In fact, in association with our findings from Sections 5.1 and 5.2, we feel encouraged that the new teaching approach conveyed the same theoretical concepts in the same depth and with the same quality as before, but did so with a clearer focus on industry challenges and with more enthusiasm from the students. The results from 2013 are consistent with this finding and indicate that the teaching approach is even more successful in conveying the course material, as suggested by the increased exam scores.

6. Discussion and Conclusion

There are some threats to validity which must be considered when interpreting the results. For example, the experiences report in Section 5.1 are subjective and may be flawed with researcher bias, as we had an interest in seeing our approach succeed. We made an effort to mitigate researcher bias by reporting on quantitative data in Section 5.2 and 5.3. Furthermore, requirements engineering is a very specific subfield of software engineering, raising the obvious question in what way the results reported herein are generalizable to other software engineering disciplines. Since the case examples had no particular focus on requirements engineering, we believe that our experiences will also hold if the milestones and student assignments were changed according to the respective needs of courses on other software engineering disciplines.
Our experience shows that students gained knowledge and awareness for the variety of problems and conflicts in project work, and developed an understanding that different contextual situations require different solutions. This allowed students to understand that it is often more important to find a solution that is not perfect but is pragmatically preferable than to find the one and only academically ideal solution. During critical course evaluation, students stressed the usefulness and helpfulness of the course with the new teaching approach. In the future, we will continue to employ industry-oriented case studies in this course and seek to extend their application to other courses, including the undergraduate counterpart of this course.

Acknowledgments

This research was partly funded by the German Federal Ministry of Education and Research (grant no. 01IS12005C). We thank Robert Pollak for proof reading the paper. Furthermore, we would like to thank Prof. Dr. Klaus Pohl for his exceptional support and the opportunity of conducting this research in his course.

References