Industry Projects in Requirements Engineering Education: Application in a University Course in the US and Comparison with Germany

Bastian Tenbergen
State University of New York at Oswego
bastian.tenbergen@oswego.edu

Marian Daun
University of Duisburg-Essen
marian.daun@paluno.uni-due.de

Abstract

Project-based learning has proven useful in software engineering education to increase student engagement and learning performance. In this paper, we contribute our experiences from applying industry projects in an undergraduate requirements engineering course in the United States. We furthermore discuss our experiences from courses conducted in Germany and the US course in light of difference in the educational systems. Results show that our course design is well received in both countries in terms of learning outcomes, student motivation, teamwork, attention to detail, and performance in the exam.

1. Introduction

Graduates of University-level software engineering programs are often hired straight out of college into industry. Competitive salaries, interesting work projects, and agile environments make going into the industry more attractive for young women and men with baccalaureate degrees in software engineering than to continue with graduate education. Therefore, there is an immediate need for undergraduate university education to produce academically and technically skilled graduates, who are familiar with cutting-edge theory just as much as they are able to work in challenging industry projects, where they are asked to produce pragmatic, profitable solutions. Yet, university education is struggling to convey an appreciation to students regarding the demands, complexity, and necessary quality required in industrial projects [18, 23]: Graduates rush to conclusions, are obsessed with the one, definitive correct solution, or aim to please the instructor, rather than exploring solution alternatives.

Hence, industry representatives often feel that university graduates require additional training before they can be useful in a company (cf. [26, 27]). Hence, there is also a need for universities to shift their instructional paradigm from traditional, theory-driven instruction [15, 26], towards more artifact-, project-, and problem-centric approaches [8, 16], where students are encouraged to actively engage in and structure their own knowledge discovery process. Evidence for the success of such a paradigm shift is paramount: students explore alternative solutions and evaluate their outcomes (see, e.g., [17]), as opposed to engage in rote memorization with poor prospects for long-term retention (see, e.g., [26]). This is especially the case for theory-heavy courses, e.g., courses on automata theory, graphs and grammars, mathematics, or requirements engineering (RE).

In previous work, we have reported on our experiences and results on student motivation and learning effectiveness from applying a course design centered around industrial case examples for undergraduate [11] and graduate [10] RE courses in Germany. The new course design was meant to combat theory-heavy instruction in favor of experiences relatable to industry. Results show that industrial case examples for student projects have a profoundly positive impact on student motivation, final exam scores, and understanding of industry challenges.

In this paper, we report on our experiences from moving the course design to an undergraduate software engineering curriculum in the United States. We will report on differences between the German and US degree programs, compare similarities and necessary changes in course design. We highlight our experiences on learning effectiveness, exam performance, and students' self-reported satisfaction. Results show that like in Germany, the new course design yields high student satisfaction, learning outcomes and teamwork. However, unlike in Germany, it was necessary to adjust content delivery in lectures to a larger degree, in part due to the vivid discussions during class meetings.

Section 2 discusses the state of the art. Section 3 reports on the German course design and summarizes experiences from its application. Section 4 illustrates the tailored course design for the US and reports application experiences. Section 5 compares the experiences and draws conclusions regarding the generalizability regarding the differences in course design and university systems. Section 5 also reports on quantified results on learning outcome and exam performance. Section 6 concludes the paper.

2. Related Work

Many approaches for industry-oriented higher education have been proposed. These often feature case

URI: https://hdl.handle.net/10125/60205
ISBN: 978-0-9981331-2-6

example-oriented instruction like in problem-based and project-based learning. (e.g., [15, 21]). Other approaches aim at bringing multiple elements from industrial practice into the classroom, but do not focus on industry case examples in particular [9], yet make extensive use of experiences gained in industry [25]. Problem-based learning has been suggested [15, 22] to improve students understanding of problem domains and to enable them deriving suitable solutions when challenged. Beside many positive reports and investigations of problem-based learning, investigations have also shown some drawbacks. For instance, there is a risk that students may become experts for a specific challenge, but not in the problem domain at large (cf. [26]). Moreover, students may not be able to adequately present their own results to other students, thereby impairing the class's overall learning success [24]. In addition, it is often perceived as challenging by instructors to ensure high course benefit and fair grading for all students [2]. Project-based learning (e.g., [8, 16]) has shown positive impact towards creativity, student enthusiasm (e.g., [1, 17]), and familiarity with problem domains at large. However, difficulty to instruct a variety of largely independent but closely related topics such that all topics are equally honored in an overarching project have been demonstrated (e.g., [5]). Also, difficulty to soundly instruct complex theoretical concepts was reported [4].

To combine project-based and problem-based learning with industry case examples, some approaches propose the involvement of real stakeholders, which is meant to allow for a realistic learning experience (see, e.g., [6, 7, 19]). In [20] real stakeholders are used to improve student motivation and engagement in building a socially relevant system for non-profit organizations. The use of non-profit organization as source for stakeholders has also been reported by other researchers (e.g., [3, 14]), as the recruitment of real stakeholder is difficult to achieve (yet, recruiting strategies for which are given in [13]). Similarly, in some approaches instructors imitate real stakeholders [28], to counter act difficulties in achieving commitment (and enthusiasm) of real stakeholders (see, e.g., [12]).

3. Applying Industrial Case Examples in RE Courses at a German University

In this section, we briefly outline the course design we used in Germany to combine problem-based and project-based learning with industry orientation. We will also briefly summarize our key findings. To foster comparability to the results from the US course, we also report on the university setting and degree program characteristics. More detailed reports on the course design and impact on graduate and undergraduate settings can be found in [10] and in [11], respectively.

3.1. Degree Program & University Setting

The course design was first applied in a graduate RE course [10] at the University of Duisburg-Essen. The graduate course is offered for elective credit for students enrolled in Business Information Systems and Applied Computer Science graduate degree programs. After the continued success of the new course design, we adopted it in an undergraduate RE course as well [11]. The undergraduate course is compulsory in both Bachelor's degree programs. The course is offered for elective credit to some non-IT majors as well.

For historic reasons, the aforementioned graduate and undergraduate degree programs are housed within the Faculty of Business Administration and Economics. As a result, the degree programs have a substantial focus on business and economics, requiring students to take at least one introductory-level course on business administration. Moreover, out of 14 required core courses, only two focus on instructing programming principles and implementation skills.

3.2. Course Technicalities

In Germany, typical university instruction employs a theory- and lecture-centric style. Ordinarily, lectures are held once a week, amended with once-weekly seminar style tutorial sessions. Their purpose is to practically apply the concepts instructed in the lecture, e.g., through weekly assignment sheets featuring academic examples (e.g., the infamous ATM machine or library system), which are prepared by students, graded by the instructors, and discussed in class.

This was the case for both the graduate and the undergraduate RE course prior to adoption of our new course design. Both courses took place over 15 weeks, each week offering one lecture and one tutorial session, both approximately 90 minutes in length. As is common in Germany, the course grade is determined by a single final exam. However, in order to be admitted for the exam, successful completion of the tutorial session is mandatory in both RE courses.

The graduate RE course is instructed annually in the summer semester (April till September) with between 15 and 45 students. The undergraduate RE course takes place each winter semester (October till March) with between 150 and 200 students.

3.3 Course Design

To foster student motivation and engagement in the RE courses, the courses make intensive use of realistic industrial case examples in combination with project-orientation. The case examples were created in close cooperation with industry partners in a large-scale national research project. To ensure instructional success, particular emphasis was placed on the following: Case examples…

- … were taken from a variety of industrial domains, including automotive, avionics, and industry automation;
- … represent typical software, requirements, and safety engineering challenges;
- … are understandable by non-experts;
- … involve a rigorous engineering process with strong emphasis on RE;
- … do not violate our partners' intellectual property.

Quality and appropriateness of the case examples were extensively discussed with our industry partners and experienced several revisions and improvements before being used in the new course design. The key aim of the new course design was to foster the following learning outcomes. Students shall:

- improve method competence, problem-solving skills, and industrial applicability;
- gain awareness of industrially relevant engineering challenges;
- foster an in-depth understanding of RE theory.

Course characteristics. Albeit due to university regulation and degree program requirements, the lecture could not be altered, the tutorial session was changed to incorporate the use of case examples. Resulting in the following major course design elements:

- In ***traditional lectures*** introductory and advanced theoretical concepts were taught. Students were encouraged to interject questions whenever.
- The main focus of tutorials were ***case study milestones***. The tutorial roster was divided into teams and each team was provided with a case example. The task was to create one specification for the system in the case example over the course of the semester. Several incremental milestones had to be submitted for review and critique. Critiquing occurred in plenum during the tutorial session throughout the semester. Students were encouraged to present preliminary results, and show failed attempts to discuss design choices, find improvements together, and allow all student teams to learn from alternative approaches. Students could resubmit revised milestones as often as necessary.
- ***Voluntary assignment sheets*** were loosely assigned and consisted of simple tasks about the lecture material. Assignments were discussed during the tutorial sessions, but never graded for credit.

In order to foster the knowledge discovery process, we neither provided sample solutions for the assignment sheets nor the case studies. Instead, tutorial instructors guided the discussions, provided further suggestions, and assisted when teams got stuck.

3.4. Experiences in Germany

In this section, we briefly summarize our findings from applying the course design in graduate and undergraduate RE courses at the German institution, particularly emphasizing the differences between the courses. Experiences in the graduate course include the following (see [10] for a detailed discussion):

Lively classroom discussions with a strong focus on theory. The classroom discussions moved away from technical questions regarding the assignments towards more content-centric topics. For instance, students started discussing different notational alternatives and how to avoid ambiguity.

More active student involvement. Students showed higher interest in how things would be in industry, what practical problems might evolve, and how pragmatic solutions could look like. Also, student participation increased: Instead of only about 20% of actively involved students, almost the whole class shared ideas and comments in the new setting.

High degree of voluntary work. Although assignment sheets were voluntary, almost all students asked questions pertaining thereto, particularly, going beyond the original scope of the assignments. Furthermore, students made extensive use of the ability to present preliminary solutions of future milestones to the entire class in order to receive peer feedback and instructor comments. We also observed an increased intrinsic effort in student solutions.

These findings were also made in the undergraduate course. In addition, the following experiences were made that differ from the graduate course (see [11]):

Concerns regarding the exam and its admission thereto. Students were far more concerned with the exam, how to achieve admission, and how to receive good grades. Students required constant assurance in the beginning weeks that obtaining admission was really "that easy." Also, frequently throughout the semester, students expressed concerns regarding the exam itself. Students often inquired about the types of questions and assignments on the exam and asked what would be considered an acceptable solution therein.

Students appreciated embedded system case examples. Albeit students expressed appreciation for insights into embedded software development at the end of the course, the use of embedded system case examples came quite unexpected to many of the students. In contrast to the graduate course, we felt that the domain of embedded systems was, albeit initially unfamiliar to undergraduate students, served as a rich domain to illustrate and practice concepts of RE.

Teamwork and participation. In the graduate course, all students applied themselves with enthusiasm and contributed to the team's progress. In the undergraduate course, this was true for many, but not all students. About half the teams complained at least one student who would not apply themselves.

Nevertheless, an increase of actively engaged students compared to prior years was noticeable.

Stronger focus on knowledge discovery. Considering that undergraduate education to some degree requires more instructor guidance than graduate education, we were surprised by the active involvement of students and their level of inquisition to related topics. Often undergraduate courses can be overly school-like, but our case example-based instruction allowed self-directed knowledge discovery. In the graduate course, the focus was on alternative solutions, in the undergraduate course on how topics fit together.

4. Application in a US-based RE Course

The opportunity presented itself to establish an undergraduate RE course at the State University of New York at Oswego (SUNY Oswego) in the US after one of the original authors changed affiliation thereto. In this section, we present the program and course characteristics following the outline from Section 3 to allow the reader to compare the two more easily.

4.1. Degree Program & University Setting

A software engineering bachelor of science (SE BS) program had been established at the institution several years prior, as part of the plan to increase the region's competitiveness in STEM. Housed within the Department of Computer Science, the SE BS program is overseen by the institution's College of Liberal Arts and Sciences. Other majors offered within the same department include baccalaureate degrees in Information Science, Computer Science, Cognitive Science, graduate programs in Biomedical Health Informatics and Human Computer Interaction, as well as minors in Information, Cognitive, and Computer Sciences. It is very common for students to switch majors in their first year (e.g., after they have identified a more suitable program within their area of interest) or in their final year (e.g., to graduate sooner and accept a job offer). To ease transition between degree programs (and hence increase retention and academic success), the core requirements of all degree programs are largely overlapping and, in contrast to the German institution, place strong emphasis on programming principles, skill acquisition, theory instruction, and computational foundations. It is hence necessary for most students to take three levels of programming courses as well as a computational theory course before advancing to elective courses. In addition, all students must satisfy general education requirements.

Since inception of the SE BS degree program, the aim was to achieve ABET[1] accreditation. To this end, several program innovations had to be enacted, one of which was the adoption of a course on software and safety RE. This new course was designed to become part of a "middleware track" in the software engineering degree program, which is a choice of three elective courses centered around software-intensive system development. However, the software and safety RE course can be taken for elective credit for all other department students.

4.2. Course Technicalities

The university emphasizes a balance between rigorous theory with practical skill acquisition. Hence, project-orientation is the key instructional paradigm at this institution, rather than lecture-heavy instruction.

Classes typically meet either twice weekly for 80 minutes or three times a week for 55 minutes for 15 weeks during the academic semester. Each class meeting can be dedicated to lecturing, class assignments, project work, or other instructional methods, at the instructor's discretion.

This was the case for the software and safety RE course. Therefore, a simple translation of the course design from the German institution was inapplicable. The course had to be redesigned in terms of topic coverage, due dates, content delivery, and course material to be appropriate for three class meetings per week over 15 weeks.

Furthermore, it is to note that in contrast to German students, American students are used to less exam-heavy evaluation and a wider range of grade-determining work products. For example, according to German university law, if a student work product factors into the student's final grade of the course, the work product becomes an "examination product" and special regulations regarding grading and archiving apply. Therefore, it is common practice to treat student work products during the semester as preamble for a final grade determining exam. The instructor of the software and safety RE course experimented in Fall 2015 in a course on software quality assurance with this paradigm, but soon found that if the final grade depends on one exam alone, the level of anxiety for each student towards the end of the course was significant. Albeit not a single student complained about this practice (neither officially, nor unofficially, nor anonymously as part of the course evaluations), casual conversations between the instructor and the students revealed that this practice translates poorly.

[1] ABET is a non-governmental organization accrediting engineering degree programs in the US, see http://www.abet.org

The course was initially offered in the spring semester (February to May) in 2017 and is instructed annually, enrolling approximately 20-30 students, typically in their third or fourth year of study. As per the institution's regulation, attendance is mandatory, however, at the instructor's discretion, not enforced.

4.3. Tailored Course Design

Since there was no prior burden that needed to be complied with, the software and safety RE course in the US could be designed completely new. Given the success of employing realistic industry-typical case example in a project-oriented fashion, this paradigm was adopted. However, considering the differences in hosting degree programs and institutional characteristics, course materials had to be redesigned, yet using the experiences from Germany. Again, the key focus was on realistic industrial case examples.

Case examples. Like in the German courses, the case examples were the most significant component of the course. The same key properties and, in part, case study descriptions from the German courses (as much as permissible as per disclosure agreements and externally available information material) were applied in the US-based software and safety RE course. Case examples included, but were not limited to:

- an airborne collision avoidance system;
- an automotive engine control unit;
- an automotive key FOB locking system;
- an adaptive cruise control system;
- a saltwater desalination plant; and
- a set of autonomous maze-escaping robots.

Course topics and learning outcomes. The key focus of the course was to instruct the principles of requirements engineering for safety-critical systems. Topics included, but where not limited to requirements elicitation and documentation using natural-language and visual languages; goal- and scenario-oriented RE; documentation of static-structural, functional, behavioral, and contextual requirements; safety engineering foundations and lifecycle; safety argumentation; hazard, risk, and safety analyses.

Learning outcomes upon completion of this course are the same as in Germany. For ABET accreditation, these have been formulated as:

- demonstrate in-depth understanding of the different types of requirements and types of requirements artifacts;
- elicit requirements for hypothetical systems under development;
- document requirements in various specification formats, including specifically diagrammatic representations at various levels of abstraction;
- differentiate "good" from "poor" requirements;
- conduct relevant analyses to detect and rectify defects in requirements impairing the safety and functional adequacy of a system;
- think abstractly about system functionality and its impact on development as well as safety;
- articulate the advantages, disadvantages, and engineering results given a problem scope, engineering results to developers, managers, and other stakeholders.

Course characteristics. To achieve the learning outcomes, the course was structured as follows:

- ***Traditional lectures*** served to convey the principles, concepts, and theories of RE. Lectures materials were very comprehensive and consisted of lecture slides as well as miscellaneous reading materials such as excerpts from text books, academic articles, and tutorials. Material presentation was agile and focused on concepts and relationships, intertwined with best practices suggestions. Lecture material was made available to all students for reference.
- Depending on class progress, five to eight ***graded biweekly assignment sheets*** were prepared in teams of two students and graded by the instructor. The aim was to get first hands-on experience with diagram notations, relationships between artifacts, purpose and meaning of concepts discussed in class, etc. An effort was made not to place emphasis on rote memorization. Assignment sheet tasks hence fostered production of artifacts and thereby understanding of the material, rather than asking questions, the answer to which can be looked up easily. Hence, grading focused on syntactic correctness and appropriateness of concepts.
- Four ***comprehensive and consecutive project milestones***, in which students were asked to produce an ISO/IEC/IEEE 29148-compliant requirements specification, enhanced with hazard and safety analyses, as well as (depending on course progress) an implementation. The purpose of the milestones was to try out the concepts and techniques within the context of a rather substantial, semester-long case example. In consequence, grading focused on applied rigor, consistency of artifacts and milestones, the system properties as specified in the specification, and overall quality. Project milestones were prepared in teams of four to six students.
- A ***midterm and a final exam*** with questions and tasks focused on documentation and analysis techniques as well as theoretical concepts as a measure of understanding of relationships between concepts and techniques.
- Additional ***voluntary assignments*** were handed out at times for preparation to provide additional practice opportunities for complicated class material. These were not graded, but occasionally discussed in class (in part, upon student request).

Class meetings were dedicated to lectures, assignment sheet discussion, project milestone presentation and discussion, or occasionally to project preparation. A strong emphasis was placed on dynamicity. Albeit the instructor maintained a semester schedule with due dates of content delivery, assignment sheets, and project milestones, these were frequently moved in light of student progress.

4.4. Experiences in the United States

In this section, we report our experiences from applying the course design at the US institution over the course of two years (Spring 2017 and 2018). In particular, we emphasize findings due to the course and institution characteristics, before we compare our findings to the Germany-based courses in Section 5.

Experiences made in the software and safety RE course in the US can be categorized into two areas: Experiences regarding class proceedings and experiences regarding student outcomes. We first report on experiences regarding class proceedings:

Lively discussion with strong focus on practicality. Like it is typical for courses at the US institution, project-based classes come with a high degree of dynamicity, as students seek access to the instructors outside of class to clarify lecture material or assignments. In the software and safety RE course, this dynamicity extended into the class meetings. Often, lively discussions were sparked during class times the instructor had initially reserved for lecturing. Students asked questions frequently about the relationship between discussed concepts and how the information translates to the assignment sheets, to the projects, and, interestingly, to real-world applications. This included students spontaneously sharing their approach to a certain problem with the rest of the class. At the midpoint of the semester, this typically evolved into an atmosphere of identifying best practices for immediate application in the project milestones and critiquing the work of others in a constructive manner.

Teamwork and eagerness to engage in class proceedings. The aforementioned lively discussion culture that naturally evolved in the course resulted in a very high degree of engagement in almost all students. With only few and occasional exceptions, all students attended class meetings and team meetings outside of class regularly, made themselves available for teamwork, and engaged in class discussions. Albeit some students were (of course) more engaged and eager to share than others, the overall degree of attentiveness experienced by the instructor was higher than in the instructor's other courses. Only two students (one in each semester), had to receive special motivation to contribute to the course.

Need for dynamic adjustment of class content, depending on semester progress. Since students naturally maintained an atmosphere of open discussion and asking questions, lectures were at times not finished on time. This means that topics often had to be taken up the next class meeting, since due to class discussions, the topic allotted for a certain class meeting was not finished. This resulted in the need to restructure the course almost on a weekly basis, adjust due dates and topic orders. In Spring 2018, class progress in early stages even motivated the instructor to change the entire semester schedule, reordering advanced topics on safety analyses and risk assessment to occur earlier in the semester. Moreover, some advanced concepts were at times easily understood by the students and lectures ended early, such that students were allowed to use class time for their project milestones, with the instructor present in the room for questions, idea exchange, and solution approach exploration. In fact, class meetings were frequently dedicated to knowledge discovery in this matter.

Experiences pertaining to student outcomes are:

Reduction in students' preoccupation with solutions desired by the instructor. Especially in the beginning stages of the semester, students were very preoccupied with instructor-desired solutions, as has been the case in the German undergraduate course. An often-uttered questions was "How do you want this to be done?" or "Is this what you wanted?" The canned response given to the students in these situations was that the *question* was inappropriate. Instead, it was emphasized that mastery of the RE process depends not on the mechanical and mindless application of some technique, but on the discovery of new information about the problem space and the system under development. It was pointed out that any way of applying a technique, any way of documenting results, and possibly, the choice to do something entirely different (even though it may violate the assignment or project instructions) may be permissible, if argued appropriately. This idea took students some time to get used to, however, all students eventually adjusted. Some students adjusted so well that they began to occasionally challenge the instructor's ideas, which occasionally sparked heated, but fruitful discussions.

At this point, it must be noted that a grading rubric for assignment sheets, project milestones, and exams were prepared by the instructor and fairly applied, yet accounted for a variety of possible solution avenues. Oral and written feedback to students critiqued both correct and incorrect solutions and always incorporated an explanation of possibly preferable ways to maximize learning benefit.

Steep learning curve regarding safety-critical systems. As outlined above, industrial case examples involved mainly safety-critical systems, in partial satisfaction of the course's purpose in the institutions degree program. Like in the German undergraduate course, since safety-critical systems are typically embedded systems, some students required some adjustment time to think in terms of the system's interaction with the operational context (outside of human users). In fact, most academic examples students were exposed to were traditional information system examples and service-oriented systems. In both semesters, it was hence necessary to dedicate class time to embedded systems.

Growing appreciation of the need to maintain consistency and quality in produced artifacts. Student artifacts in the beginning of the semester were typically vague, abstract, and superficial. Often, student teams would procrastinate work until the due date and produce artifacts in corresponding quality. This behavior became gradually less frequent as students realized the importance of diligence and consistency between artifacts. One student indicated that maintaining consistency and the appropriate level of detail across diagrams and natural language requirements quickly became a semester-dominating task, yet he felt it was necessary to deliver the case study in the appropriate level of quality. One recurring question, however, was about the appropriate level of detail in diagrams, especially regarding the number (and naming) of modeling elements therein. Rough instructor-provided guidelines helped to some degree but left some students with uncertainty about the quality of their own work.

5. Comparison of Experiences and Results

In the following, we highlight differences in course setups and highlight the student learning outcomes in Germany and in the US.

5.1. Differences in Course Setups

Due to differences in universities and degree programs, there are some subtle differences between the courses. Table 1 highlights the main differences between the two courses in either country. While most differences had only a minor impact (e.g., the instructional time was roughly 180mins per week, see Table 1), experiences may have been impacted by the peculiarities of the course designs. Below, we discuss the specifics, which may have impacted our findings the most. We compare our experiences in Section 5.2.

Determination of the final grade. The courses taught in Germany determined the final grade on the basis of just one final exam, as is the norm. The project milestones served as an ungraded precondition for exam admission. In the US, the final grade depended on the overall, equally weighed score between assignment sheets, project milestones, midterm and final exam.

Topics covered. The German undergraduate course mainly focusses on core RE activities like elicitation, negotiation and documentation of requirements. Emphasis is particularly given to the model-based specification of requirements, context analysis and structured system analysis. The graduate course focusses on advanced concepts of requirements engineering. Particular emphasis is given to goal-/scenario-oriented RE and essential system analysis. In the US, topics like structured analysis and essential system analysis were dropped in favor of topics pertaining to safety engineering, safety argumentation, hazard analyses, safety analyses, and safety lifecycles.

Table 1 Comparison of the Differences in Germany and in the US

	Germany	US
Student Body	Applied Informatics: Sys Engineering Business Information Sys	Computer Science Information Science Software Engineering
Degree Program Foci	business admin., software eng. processes	technical skills, softw.-intensive systems
Number of Students	15–45 (graduate)	20–30
	150–200 (undergraduate)	
Course Meetings	weekly lectures & tutorials (90min each); tutorials were dedicated to assignments, projects, or discussions	three weekly meetings (55mins), dedicated to lecturing, assignments, projects, or discussions (instructor's discretion)
Student Evaluation	single final exam, 100% of final grade (admittance to exam depends on completion of tutorials)	project milestones, assignment sheets, midterm & final exams, each 25% of final grade
Course Topics	goal and scenario-oriented RE; essential sys analysis; reqs validation & management (graduate)	same as both, however without structured and essential sys analysis. Additionally: safety engineering foundations and lifecycle; safety argumentation; hazard, and safety analyses; risk assessment
	Requirements elicitation & documentation; specification with natural and graphical languages; requirements perspectives and system's context; structured sys analysis (undergraduate)	
Lecture Content Delivery	strictly according to semester plan	frequent changes depending on project progress
Assignment Sheets	voluntary	mandatory

Course materials and student prerequisites. All course materials were prepared specifically for their respective course. In particular, teaching materials for the US-based course have been prepared specific to the needs of the degree programs. For example, while in the US, a course on software engineering methodology using UML is a prerequisite for the software and safety RE course, such a course was not required in Germany. Hence, in the US, instruction could focus on RE-specific concepts and usage of diagrams rather than on their notation. Yet, the student prerequisites in Germany and the US were comparable, because the degree programs from which students were recruited have a focus on software engineering processes.

In all other respects, the courses were similar in theoretic depth, level of detail, and covered topics.

5.2. Comparison of Experiences

Albeit subtle differences (see Section 5.1) exist between the courses in Germany and in the US, which to some degree limit comparability between the courses, there are several corresponding observations:

Teamwork and Enthusiasm. In both courses in Germany as well as in the US course, we noted a large degree of student enthusiasm as well as teamwork. In all three courses, students were informed that the underlying assumption is that all case study artifacts are created by all team members to an equal degree, the reality is that often, students segregate their work and only work on parts of the assignment. Project-based education is a stable in the department housing the US course, and unequal contributions to project milestones is a chief complaint of students. Yet, in our experience, after a few weeks in the US course, this behavior changed in almost all teams towards a truly cooperative environment, where students sought collaboration and discussions about the case study. We observed the same in both German courses.

Grade preoccupation and voluntary work. We have discussed students' preoccupation with grades and suggested that the project-based approach mainly holds value for industry-readiness instead of improving academic achievements [11]. This finding was not replicated with the undergraduate students in the US course. Students rarely inquired about the midterm or final exam and trusted that the exams were similar in structure and difficulty as the graded assignment sheets and case study milestones. In consequence, albeit a similar amount of additional voluntary assignments was made available, no student submitted voluntary work for feedback. Nevertheless, some claimed that they at least read through these assignments. It is noteworthy, that grade preoccupation is quite common in other courses in the US department.

5.3. Student Satisfaction and Exam Scores

To complement our qualitative reports from Sections 3.4, 4.4 and 5.2, this section reports quantitative findings from the student evaluation results geared towards self-reported satisfaction with the learning outcomes as well as the exam results.

Student Satisfaction. In Germany the new course design improved the students' self-reported satisfaction with the learning process as compared to previous years as well as the department average. These results are based on course evaluations conducted regularly by the German university's teaching quality assurance program. The US institution also has such a program, however, the specific evaluation questions differ. To ensure comparability, the student evaluation in the US has therefore been extended with the questions asked in Germany. Since the course has been newly conceived at the US institution, a comparison to a previous course design is not possible. Therefore, in Fig. 1, we show results of the US course evaluation and provide results from Germany for guidance. Yet, it must be noted that due to the differences outlined in sections 3.3, 4.3, and 5.1, the comparison is tendential at best, as the differences in the courses warrant no strict comparison.

Answers were collected on five-point Likert scales, which were converted to an interval scale to allow computing the mean for each question. These means are shown in Fig. 1. Averages of previously reported results [10], [11] are shown as dashed lines with triangular markers. The students' self-reported satisfaction with the US course is shown as solid lines. As can be seen, in both semesters, US course has been rated similarly positive like the graduate and undergraduate courses in Germany. Albeit the differences between courses do not allow for a direct comparison, we take the results from Fig. 1 as positive indications that using industrial case examples are applicable in a US setting and yield a similar positive evaluation and underline our qualitative impressions (see Sections 4.4 and 5.2).

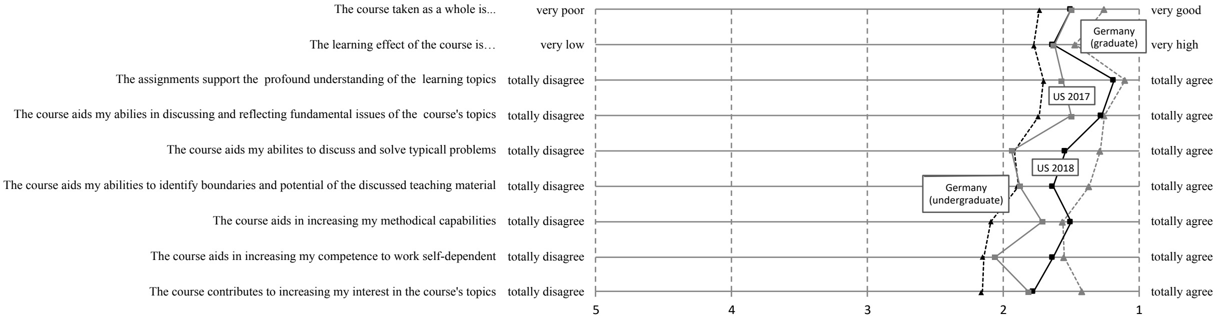

Fig. 1 Results of the Annual Student Evaluation

Exam Scores. Fig. 2 shows the student exam results as a percentage of maximum attainable points for all courses. For the German courses, average values after introduction of the new course design are given and depicted in light gray. For the US course, averages are given for midterm and final exam for both the 2017 (dark gray bars) and 2018 course (black bars). Exam tasks in both Germany and the US were similar in style, complexity, and difficulty like assignments and project milestones, however adjusted to an exam time of ca. 90 minutes (Germany) and 45 minutes (US).

In both years, midterm and final exam grades are similar to the German graduate RE course, yet final exam grade averages in the German undergraduate course are lower than in the other courses. This can be explained by the fact that in the German graduate course as well as both semester in the US, the RE course could be taken for elective credit. Hence, it can be presumed that students chose the course due to interest and thus performed better on exams. The fact that the grade ranges are rather wide in the German undergraduate course and that in both 2018 exams, the maximum grade was 100% seemingly support this conjecture. Nevertheless, it must be noted that of course, exam performance is not directly comparable. Thus, we provide these figures for reference only.

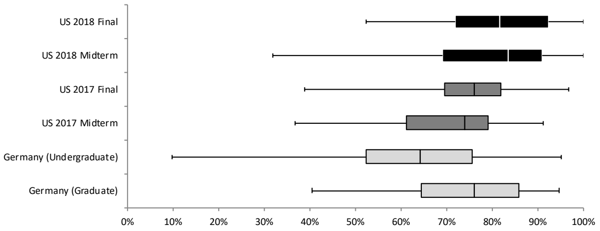

Fig. 2 Exam Results in the US compared to Germany

6. Conclusion & Outlook

In this paper, we have reported on our experiences, students' self-reported satisfaction, and exam results of a software and safety requirements engineering course in the United States that heavily focuses on the use of realistic, industry-typical case examples. We have introduced our course design, summarized prior findings from application in graduate and undergraduate courses in Germany, and reported on new results on the course design's application in the US. We have outlined differences in course design, where applicable, and highlighted resulting differences and similarities in experiences.

Our experiences show that country-specific differences in grading, student population, student prerequisites, as well as course organization and schedule made it necessary for the instructor to pay close attention to class progress and content delivery to a higher degree in the US than in Germany. Nevertheless, exam results and student evaluations underline our strong positive qualitative experience with the success of this teaching paradigm: case example-focused instruction yielded a truly cooperative learning environment characterized by active knowledge discovery, and a fruitful discussion culture in the US, as it did in Germany. We are therefore confident that this teaching approach translates well, regardless of organizational differences.

Main threats to validity include comparability of courses, instructor bias, and generalizability. To allow for easier comparison of courses (and experiences therein), we have highlighted similarities and, where applicable, differences in course design and the university system and illustrate our experiences in light thereof. Moreover, the instructors are biased in that they want their students and their teaching approach to be successful. Yet, we argue that it is just this preoccupation with student outcome that underlines the success of this course design. Of course, given the nature of this report, results are not universally generalizable. Nevertheless, we have provided details on similarities and differences in the university settings to allow the reader to evaluate the experiences and applicability of the course design in a setting that is similar to the ones mentioned herein. We have furthermore provided quantitative results as guidance for rough comparison. However, we caution the reader that strict comparison of outcomes cannot be undertaken.

7. Acknowledgements

We would like to thank Th. Weyer and K. Pohl for their collaboration on designing and conducting the course design in Germany.

8. References

[1] Alfonso, M.I., and A. Botia, "An Iterative and Agile Process Model for Teaching Software Engineering", *18th Conference on Software Engineering Education Training (CSEET'05)*, (2005), 9–16.

[2] Azer, S.A., "Introducing a problem-based learning program: 12 tips for success", *Medical Teacher 33*(10), 2011, pp. 808–813.

[3] Boehm, B., A. Egyed, D. Port, A. Shah, J. Kwan, and R. Madachy, "A stakeholder win–win approach to software engineering education", *Annals of Software Engineering 6*(1), 1998, pp. 295–321.

[4] Bonwell, C.C., and J.A. Eison, *Active Learning: Creating Excitement in the Classroom*, Jossey-Bass, Washington, DC, 1991.

[5] Boss, S., and J. Krauss, *Reinventing Project-based Learning: Your Field Guide to Real-world Projects in the Digital Age*, International Society for Technology in Education, 2007.

[6] Bruegge, B., S. Krusche, and L. Alperowitz, "Software Engineering Project Courses with Industrial Clients", *Trans. Comput. Educ. 15*(4), 2015, pp. 17:1–17:31.

[7] Brügge, B., and M. Gluchow, "Towards production ready software in project courses with real clients", *2012 First International Workshop on Software Engineering Education Based on Real-World Experiences (EduRex)*, (2012), 5–8.

[8] Burge, J.E., "Application and Appreciation: Changing Course Structure to Change Student Attitudes", *2009 22nd Conference on Software Engineering Education and Training*, (2009), 45–52.

[9] Dagnino, A., "Increasing the effectiveness of teaching software engineering: A University and industry partnership", *2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE T)*, (2014), 49–54.

[10] Daun, M., A. Salmon, B. Tenbergen, T. Weyer, and K. Pohl, "Industrial case studies in graduate requirements engineering courses: The impact on student motivation", *27th IEEE Conference on Software Engineering Education and Training, CSEE&T 2014, Klagenfurt, Austria, April 23-25, 2014*, IEEE (2014), 3–12.

[11] Daun, M., A. Salmon, T. Weyer, K. Pohl, and B. Tenbergen, "Project-Based Learning with Examples from Industry in University Courses: An Experience Report from an Undergraduate Requirements Engineering Course", *29th IEEE International Conference on Software Engineering Education and Training, CSEET 2016, Dallas, TX, USA, April 5-6, 2016*, IEEE (2016), 184–193.

[12] Gabrysiak, G., H. Giese, and A. Seibel, "Why should I help you to teach requirements engineering?", *2011 6th International Workshop on Requirements Engineering Education and Training*, (2011), 9–13.

[13] Gabrysiak, G., M. Guentert, R. Hebig, and H. Giese, "Teaching requirements engineering with authentic stakeholders: Towards a scalable course setting", *2012 First International Workshop on Software Engineering Education Based on Real-World Experiences (EduRex)*, (2012), 1–4.

[14] Gabrysiak, G., R. Hebig, L. Pirl, and H. Giese, "Cooperating with a non-governmental organization to teach gathering and implementation of requirements", *2013 26th International Conference on Software Engineering Education and Training (CSEE T)*, (2013), 11–20.

[15] Garg, K., and V. Varma, "A Study of the Effectiveness of Case Study Approach in Software Engineering Education", *20th Conference on Software Engineering Education Training (CSEET'07)*, (2007), 309–316.

[16] Li, Q., and B.W. Boehm, "Making winners for both education and research: Verification and validation process improvement practice in a software engineering course", *2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE T)*, (2011), 304–313.

[17] Meawad, F., "The virtual agile enterprise: Making the most of a software engineering course", *2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE T)*, (2011), 324–332.

[18] Oh, E., and A. van der Hoek, "Towards game-based simulation as a method of teaching software engineering", *32nd Annual Frontiers in Education*, (2002), S2G-13 vols.3-.

[19] Penzenstadler, B., M. Mahaux, and P. Heymans, "University meets industry: Calling in real stakeholders", *2013 26th International Conference on Software Engineering Education and Training (CSEE T)*, (2013), 1–10.

[20] Penzenstadler, B., D. Richardson, B. Karlin, A. Cook, D. Callele, and K. Wnuk, "Using Non-Profit Partners to Engage Students in RE", pp. 10.

[21] Richardson, I., and Y. Delaney, "Problem Based Learning in the Software Engineering Classroom", *2009 22nd Conference on Software Engineering Education and Training*, (2009), 174–181.

[22] Santos, S.C. dos, "PBL-SEE: An Authentic Assessment Model for PBL-Based Software Engineering Education", *IEEE Transactions on Education 60*(2), 2017, pp. 120–126.

[23] Shaw, M., *Software Engineering for the 21st Century: A basis for rethinking the curriculum*, Carnegie Mellon University, Pittsburgh, PA, 2005.

[24] Sweller, J., "The worked example effect and human cognition", *Learning and Instruction 16*(2), 2006, pp. 165–169.

[25] Tahmoush, D., S. Fouché, S. McMaster, and J. Purtilo, "Enhancing Software Project Management Courses With Industry Participation", *FECS'09 - The 2009 International Conference on Frontiers in Education: Computer Science and Computer Engineering*, , pp. 6.

[26] Varma, V., and K. Garg, "Case studies: the potential teaching instruments for software engineering education", *Fifth International Conference on Quality Software (QSIC'05)*, (2005), 279–284.

[27] Wohlin, C., and B. Regnell, "Achieving industrial relevance in software engineering education", *Proceedings 12th Conference on Software Engineering Education and Training (Cat. No.PR00131)*, (1999), 16–25.

[28] Zowghi, D., and S. Paryani, "Teaching requirements engineering through role playing: lessons learnt", *Proceedings. 11th IEEE International Requirements Engineering Conference, 2003.*, (2003), 233–241.